

令和5年度姫路市大学発まちづくり研究助成事業

# 動画を用いた機械学習による行動解析

令和6年3月

兵庫県立大学 環境人間学部  
画像メディア研究室  
(研究代表者：木村敏文)



## 目次

第1章 研究背景	1
第2章 機械学習を用いた行動解析支援システム開発	3
第3章 動画を用いた機械学習による行動解析	12
第4章 まとめ	18
参考文献	19
謝辞	20

## 付録

- 行動解析支援システム・環境構築ガイド（簡易版）
- 行動解析支援システム・環境構築ガイド（詳細版）



## 第1章 研究背景

### 1-1. 研究背景

姫路市のみならず、イベントの効果検証や様々な計画の基礎資料として、人数等をカウントする調査が行われることが多い。イベントや街中における人の人数を把握すること、人流を解析することは、行政における計画の元となる基本かつ貴重なデータである。これらのデータを得るためには現場における人海戦術もしくは記録映像を用いたカウントなどを強いられ、多くの時間を要する辛い作業を行うことが現在でも最も有効な手段の1つである。しかし、働き手の確保が困難であること、人件費の高騰から財政圧迫、調査中止の判断等、様々な課題を抱えている。このことから、コンピュータによる支援が望まれている。

現在、多くの自治体でコンピュータやAIを活用した取り組みがなされている。平成30年に国土交通省から出された「スマート・プランニング実践の手引き」(国土交通省, 2018)には、個人単位の行動データに基づく新たなまちづくり、というサブタイトルが付いており、岡山市や神戸市におけるGPSやWi-Fiを活用した行動解析の例が挙げられている。多くのデータを収集できることは確認できているが、データスクリーニングが必要であることが示唆されている。近隣の都市において、加古川市はスマートシティの取り組みとして見守りカメラを設置している(多田, 2022)。カメラの映像と共に、ICタグを用いた小学生や高齢者の見守りも行っている。他にもデータ利活用を進めるべく、若者の意見もとりにこむような住民参加ツールも活用している。AIを用いた取り組みも進めている。今後、個人情報を考慮しつつ、このようなビックデータをどのようにデータを活用していくのか、ということが課題となっている。

本研究では、ビックデータの活用の一環として、動画から把握できる基本的な情報(人数、人の集中箇所等)を機械学習の技術を用いて、自動で処理する手法を提案することを目的とする。この問題を解決するための1つの方法として、動画を用いた機械学習による行動解析システムの導入が挙げられる。我々の研究グループでは情報技術を用いることで、職員の方々が手軽に人のカウントや流れを把握できるシステムの開発を目指した研究を行った。

最初の段階として、人の検出が必要である。Redmonらが機械学習の技術を用いたリアルタイムオブジェクト検出アルゴリズムとしてYOLO(You Only Look Once)を開発している(Redmon et al., 2016)。YOLOは画像を深層ニューラルネットワークに1度処理する間にオブジェクトの検出とクラス分類、2つのプロセスを同時に行うことで高速なオブジェクト検出を実現している。事前に学習したモデルを用いて、検出対象オブジェクトを高精度で検出することが可能である。YOLOはこれまでにアップデートを繰り返し、一般的なオブジェクト(人も含む)について高精度で検出できる。YOLOには時間的な関係を把握する仕組みが実装されていないため、研究代表者である木村らが開発した昆虫を対象とした複数個体同時追跡システムK-Trackの個体識別手法を組み合わせ、移動軌跡描画や位置情報取得を試みる(Kimura et al., 2014, 2019)。

動画から検出対象の識別が可能になることで行動解析を行うための基礎データ取得が可能となる。毎年繰り返し行う交通量調査やイベントの人数カウント等の業務の省力化に繋がると共に全国的に取り組まれているスマートシティの実現にも寄与することが期待できる。業務効率化、省力化に寄与する研究であるため、姫路市のみではなく、広く応用が可能であると考えられる。本研究の最終目標としては、姫路市の職員が自ら解析することができ、業務に活用することである。

## 1-2. 研究のスケジュール

本研究は以下のスケジュールで実施した。

日程	実施内容
2023/4/10	研究助成申請打ち合わせ@兵庫県立大学・画像メディア研究室
2023/4/28	事前審査申請書の提出
2023/6/5	審査結果の通知
2023/7/3	連携課との顔合わせ会@姫路市役所
2023/7/26	研究打ち合わせ@姫路市役所
2023/8/26	姫路駅前調査（動画撮影）@姫路駅前
2023/10/30	中間発表用スライドの提出
2023/11/8	姫路市大学発まちづくり研究助成事業中間発表会 @姫路市総合福祉会館
2023/12/15	研究打ち合わせ@兵庫県立大学・画像メディア研究室
2024/1/15	加古川市・企画部 政策企画課へのヒヤリング調査@加古川市役所
2024/2/9	解析システム操作体験会@兵庫県立大学・画像メディア研究室
2024/3/11	成果報告書データおよび発表用スライドデータ提出
2024/3/28	姫路市大学発まちづくり研究助成事業成果発表会 @姫路市役所北別館 成果報告書（冊子版）の提出

## 第2章 機械学習を用いた行動解析支援システム開発

### 2-1. 行動解析支援システムの設計

令和5年度姫路市大学発まちづくり研究助成事業申請にあたり、兵庫県立大学・姫路環境人間キャンパスにて、姫路市都市局まちづくり部都市計画課・岩崎氏、有方氏、環境人間学部・教授 太田教授、と、事前打ち合わせを行った。最初に有方氏から都市計画課としての計画について、説明があった。都市計画課では、「居心地が良く歩きたくなるまちなか」を目指し、2022年9月から姫路駅前において、ピオレ主催の社会実験を行っている。姫路駅前広場にベンチ・カウンタを設置し滞留できる場所を作る計画である（図1）。社会実験の概要は以下のとおりである。

開催日時： 2022年9月～ 現在も継続中

場所： 姫路駅北口駅前広場（ピオレ前）

主催： ピオレ姫路（アーバン開発 /JR 西日本 スターバックス等）

開催支援： 姫路市等

実施内容： 広場空間にベンチ及びテーブルを設置

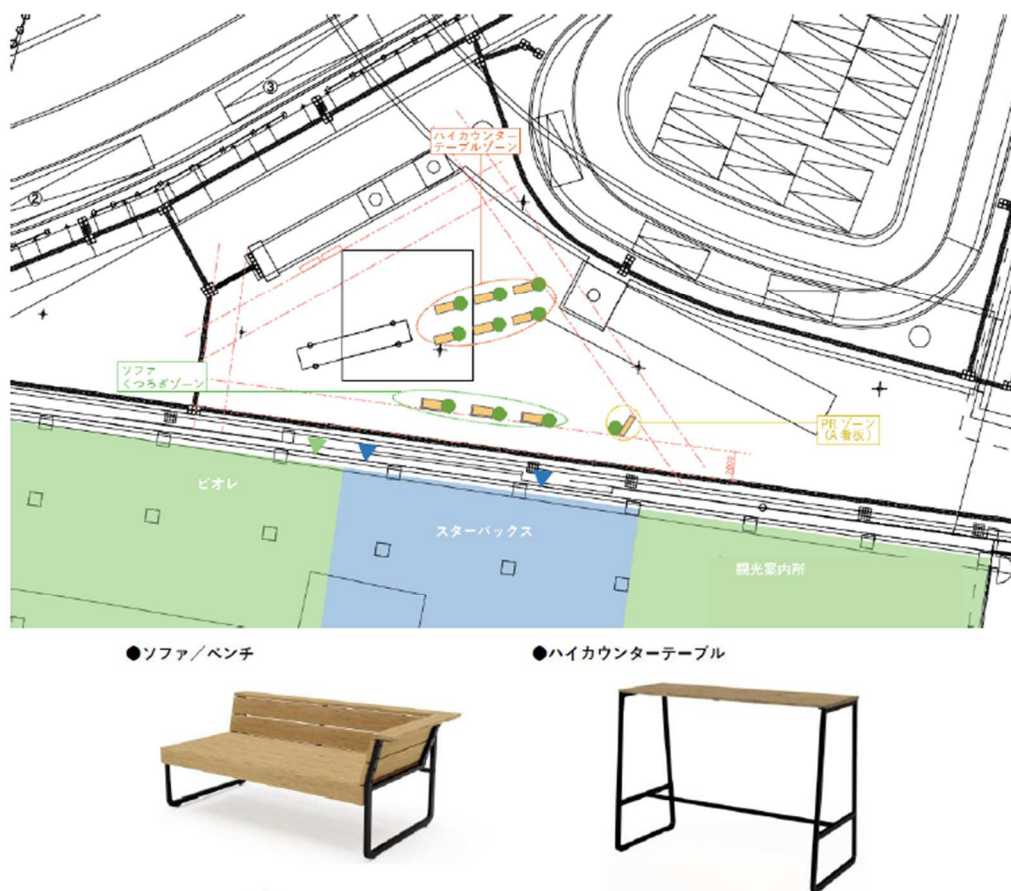


図1. 社会実験におけるソファベンチおよびハイカウンターテーブルの設置場所

姫路市における社会実験の効果検証は、人が直接行う原始的な調査（効果測定や通行量調査）を実施していることが多く、調査に人件費が掛かることや目視による限界、日程調整が困難なことが課題である。社会実験の効果検証方法として、AI等の新たな技術の活用について、検討を行っていた。「動画を用いた機械学習による行動解析」についての可能性について、検討を行った。本研究は機械学習といった情報技術を用いた行動解析支援システムを開発することと共に、「姫路市職員が自ら解析し、業務に活用できるようになる」が目標のため、可能な限り、使いやすいシステムを目指すことを念頭においた設計を心がけた。工学系、情報工学系が得意な職員のみならず、文系出身、情報工学系が苦手な職員にも受け入れられるようなシステム（手法、手順）設計を目指した。

行動解析で行える調査項目や撮影条件等の課題を整理し、姫路市におけるどのような業務への実装が可能かを把握していくところから始めた（図2）。姫路市ではイベント時や日常における人の流れの基礎資料として作成するために、アルバイトを雇用し、人数等をカウントする調査を多く行っている。このような調査は、人口減少による働き手の確保が困難になることや、人件費の高騰から財政圧迫、調査中止の判断等、様々な課題が考えられるため、このような作業をコンピュータで支援することが今後の行政サービス向上に必須である、と考える。

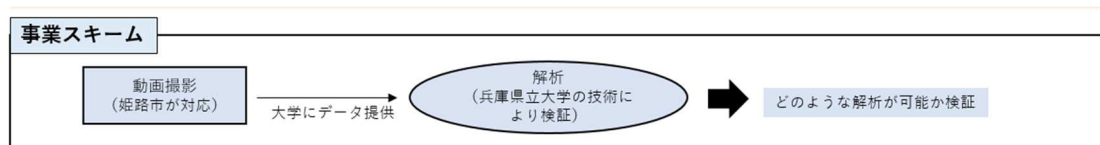


図2 本研究における事業スキーム

検討していく中で、調査したい項目として、「人数のカウント」「集中箇所の把握」「移動速度の把握」「滞在・停止時間の把握」「属性（年齢・性別等）の把握」「行動内容（飲食・会話等）の把握」「意識・意向の把握」を挙げ、作業の複雑さを基に分類を行った（図3）。

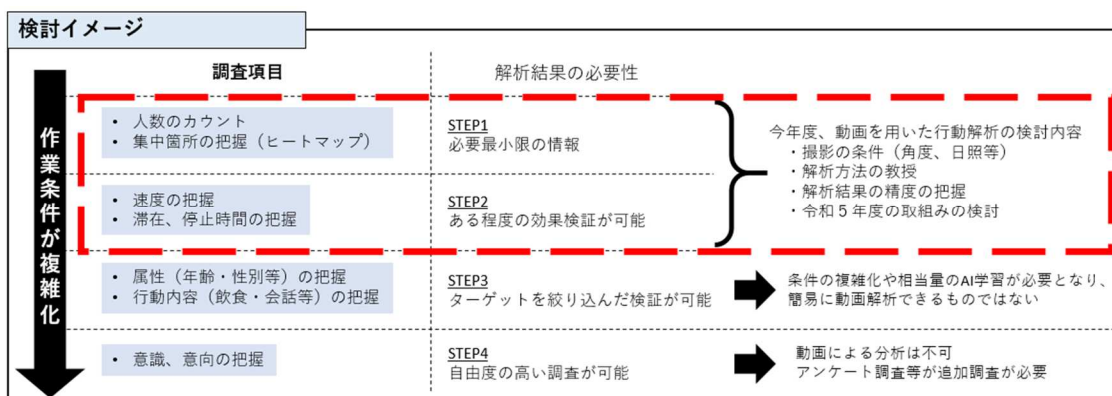


図3 本研究における調査項目と解析の必要性



作業の複雑さに応じて、STEP 1～4を割り当て、本研究において行う課題について、検討を行った。本研究で利用する機械学習のアルゴリズムは Redmon らが提案しているリアルタイムオブジェクト検出アルゴリズム YOLO(You Only Look Once) である (Redmon et al., 2016)。YOLO は深層ニューラルネットワークに画像を1度処理する間に、2つのプロセス (オブジェクト検出とクラス分類) を同時に行うことで高速なオブジェクト検出を実現している。画像1枚あたり0.01～1秒程度 (検出オブジェクトの数やPCの性能によって検出時間は変わる) で処理が可能のため、多様な用途での活用が期待できる。また、YOLO はアップデートを繰り返しており、性能は向上している。コマンドプロンプトからコマンドを入力する、という慣れない作業を伴うが、できるだけ単純な作業とできるようにした。

機械学習の技術を使うとはいえ、万能ではないので、現状でできること、できないことを整理することと共に、姫路市職員の方に日々の業務で活用できるように、できるだけ簡単な作業で実現できそうなものをピックアップしていった。調査項目と解析の必要性 (図3) について、STEP 3 および 4 は1年で行う課題としては大きいため、今後検討していく課題として、位置づけた。STEP 1 および 2 は令和5年度で検討していく課題としては良いサイズであると考え、この2つのSTEPについて実現を目指した。

STEP 1, 2 から考えられる調査項目を図4に示す。令和5年度に検討を行う課題としては、(1) 通行量調査、(2) ベンチの活用人数調査、(3) 集中箇所の把握 (ヒートマップ) を挙げる。

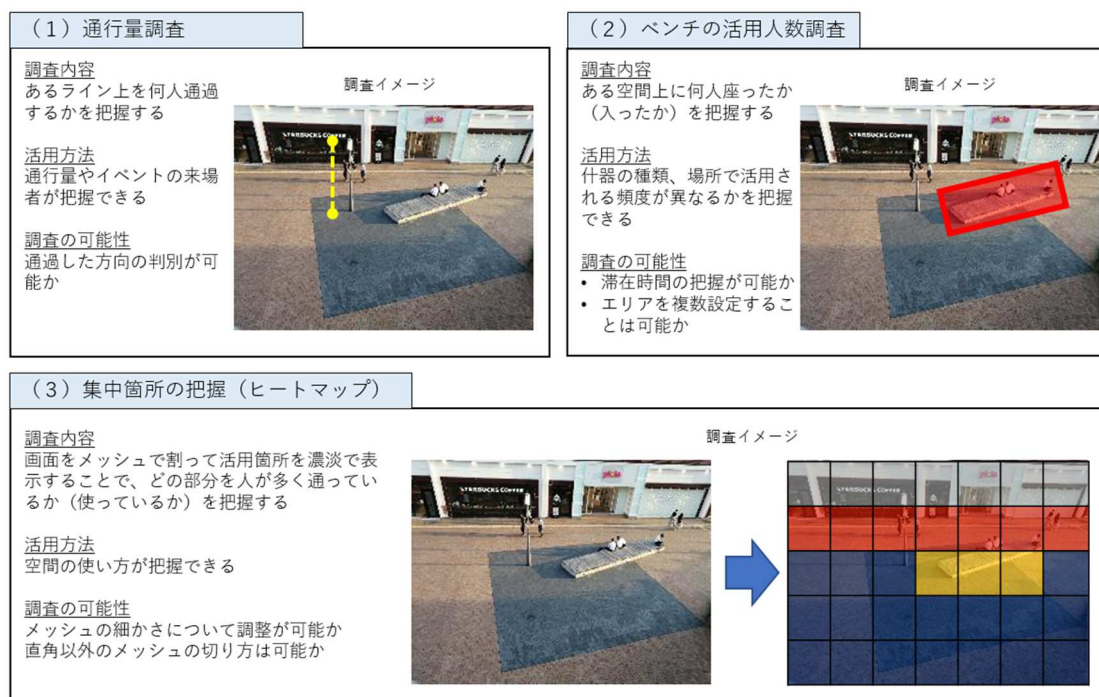


図4 本研究における課題設定

3つの課題を実現するために動画から取得する必要がある共通データは検出対象ごとの位置情報である。日本で使われている動画の仕様では1秒間を30枚の画像で表現している。各画像における検出対象を紐づけることができれば、時空間的な情報を得ることが可能となる。このことを踏まえて、3つの課題に共通する課題として、1) 検出対象決定、と、2) 位置情報の取得、を挙げる。1つ1つの課題について、必要な情報を考える。

(1) 通行量調査、については、検出対象を識別することで実現可能である。画像ごとの位置情報、画像間の関係を紐づけることによって、可能である、と考えられるため、研究代表者である木村がこれまで行ってきた平面アリーナ上で行動する複数ミツバチ同時追跡システム K-Track で用いた手法 (Kimura et al., 2014) を活用することで可能性を検討する。

(2) ベンチの活用人数調査、についても、(1)と同様な考え方で実現できると考えられるため、(1)と合わせて、検討していくこと、とした。

(3) 集中箇所の把握 (ヒートマップ)、については、(1)、(2)に比べて、難易度が低い、と考える。2つの課題で必要だった検出対象の識別が必要なく、各画像における検出対象の位置情報を用いることで、実現可能である。

これらのことから、令和5年度の課題として、(3)をメインの課題とし、(1)(2)についても検討していくこと、とした。

これらの課題を実現するために行動解析支援システムに必要な共通機能は、以下の2つが挙げられる。

- ・ 検出対象の検出
- ・ 検出対象の位置情報取得

それぞれの課題も含め、機械学習を用いた行動解析支援システムを提案する。

## 2-2. 行動解析支援システムの概要

行動解析支援システムとして、次の手順で処理していく

1. 動画の撮影
2. 検出対象の検出（人数カウント）
3. 検出対象の位置情報の取得
4. ヒートマップ作成（集中箇所検知）
5. 検出対象の識別
6. 検出対象の行動追跡
7. 検出対象の行動解析（滞留解析）

以下、手順1～3は共通課題、4以降はそれぞれの課題に対する手順である。それぞれの手順について、処理方法を説明していく。

### 手順1) 動画撮影

デジタルビデオカメラ等で解析対象地域の撮影を行う。人数カウント、行動解析するためには可能な限り、同一画角で撮影することが望ましい。また、撮影中は手振れや画角の変化がないように、三脚などで固定し、撮影することが必須である。映像に手を加えず、画像処理することを考えて、撮影することを検討することが大切である。



図5 動画の撮影例

手順2) 検出対象の検出 (人数カウント)

手順3) 検出対象の位置情報の取得

検出対象を検出する手法として、ニューラルネットワークをベースとした対象検出手法 YOLO を用いる (Redmon et al., 2016)。事前に学習されたモデルファイルが提供されており、高速かつ高精度で学習済みの対象を検出することが可能である。比較的容易にプログラムの実行できることから多くの研究で利用されている。

YOLO として、YOLOv7 (Wang et al., 2023) を用いることとした。本研究では YOLOv7 で提供されている人の検出に特化した学習モデル (yolov7-w6-person.pt) を利用することで、人の検出、カウントを行う。撮影した動画を YOLOv7 を用いて、画像・映像処理し、人を検出する。YOLOv7 の処理結果として、検出画像 (図 6) および検出結果 (分類、位置情報、確率/図 7) を出力する。

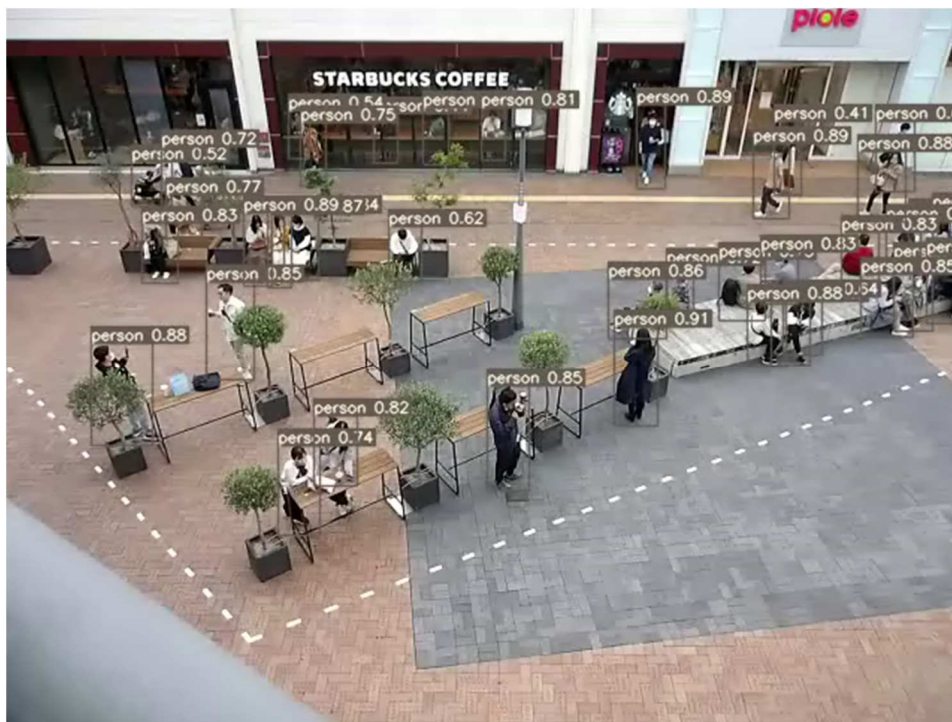


図 6 YOLOv7 による検出例 (person のあとの数字は検出対象の確率)

	A	B	C	D	E	F	G
1	0	0.576563	0.261458	0.028125	0.08125	0.789062	
2	0	0.621094	0.4125	0.0421875	0.108333	0.883789	
3	0	0.655469	0.416667	0.0453125	0.0875	0.890625	
4	0	0.435156	0.285417	0.0328125	0.104167	0.900391	
5	0	0.10625	0.2875	0.05	0.116667	0.932129	
6							

A列：分類番号  
 B列：検出領域の右上・x方向の相対座標  
 C列：検出領域の右上・y方向の相対座標  
 D列：検出領域のx方向の相対的な大きさ  
 E列：検出領域のy方向の相対的な大きさ  
 F列：分類された物体である確率

} 検出物体の位置と大きさがわかる

図7 YOLOv7による検出結果（カンマ区切りのファイルとして、出力される）

YOLOは検出した物体の分類番号、検出した物体がいる領域（右上の座標（縦、横）および大きさ（縦、横幅）を検出結果として出力する。座標や大きさは画像の大きさに対する相対的な位置や大きさで表されているため、画像の大きさに依存しない検出手法と言える。この出力結果を用いて、手順4）～7）の実現を試みる。

#### 手順4）ヒートマップ作成（集中箇所検知）

手順4）から7）の中で、最初の実現しそうな課題はヒートマップの作成である。手順3）で得られた位置情報を基にヒートマップ作成の手順を示す。

- (1) YOLOv7のよる人検出結果を基に位置情報を取得する
- (2) 画像を  $n \times m$  のセルに分割する（図8）
- (3) 位置情報を基に各セルに人がいるかどうかを判定する
- (4) 画像ごとに人がいる割合を計算する（単位時間当たりの割合を計算）
- (5) 画像にマッピングする

分割する数によって、得られる情報の細かさが決まる。本研究では  $15 \times 15$  の大きさを標準（図8, 図の説明では  $10 \times 10$  を使用）として、システムの開発を行っている。必要に応じて、分割する数を変えることも可能である。各セルでの人判定はYOLOv7の検出枠の中心点で判定することとした。ヒートマップは黒から赤、黄色を経て、白へと変わることによって集中している（図9）。元画像にマッピングすることにより、集中箇所を視覚的に把握することができるようになる。また、映像を画像に分解したのち、画像ごとにヒートマップを作成することから、人が集中する箇所を時間経過で確認することも可能である。



図8 ヒートマップ作成のためのセル設定 (10×10)



図9 ヒートマップの一例 (15×15)

手順5) 検出対象の識別

手順6) 検出対象の行動追跡

手順7) 検出対象の行動解析 (滞留解析)

手順5)、6)、7) を実現するためにも手順3) で得られた位置情報を活用する。画像ごとに検出した対象の識別ができれば、行動追跡も可能、となるため、これらの処理をまとめて説明する。YOLO (YOLOv7) では対象を高精度で検出できる反面、時間的な関係が考慮されていないため、画像が変わると同一の物体であることが判断できない。ここで、研究代表者である木村らが開発した K-Track の個体識別法 (Kimura et al., 2014, 2019) を用いて、検出対象の識別を行う。識別ができれば、移動軌跡を描くことが可能となり、行動追跡につながる。個体識別の手順を簡単に説明する。基本的な考えとしては、時空間的な関係を紐づける、という考え方を基に画像ごとの検出物体を紐づけていく。

(1) YOLOv7 で検出された物体 (人) ごとに検出枠と中心座標を取得する

(2) 最初の画像において、検出された物体ごとに番号を割り振る

(3) 元になる画像 (番号が割り当てられている画像) と1つ後の画像を重ねる

(4) 画像間で物体ごとに検出枠の重なり、と、中心座標間の距離を計算する

(5) 1つ後の画像における物体には重なりが大きく、中心距離が近い番号を割り当てる。割り当てる物体がない場合は新しい番号を付与する

(6) 画像ごとに (4)、(5) を繰り返し、最後の画像まで処理を進める

(7) 画像ごとに同じ番号の物体の中心点をつなぐことにより、移動軌跡を描く個体が識別できることで、それぞれの検出対象の動き (行動) が解析でき、移動速度や滞留時間の把握にもつながる。



図10 検出対象の行動追跡 (15×15)

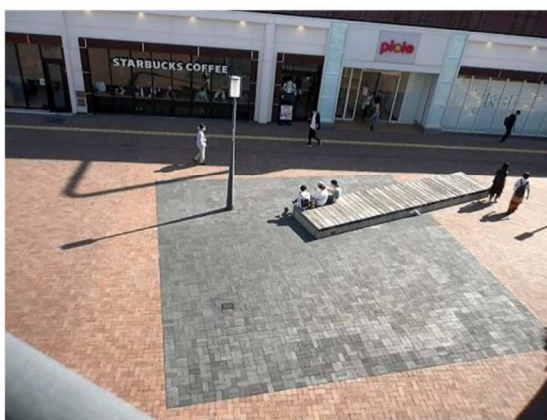
### 第3章 動画を用いた機械学習による行動解析

#### 3-1. 提案手法による行動解析実験の設定

提案手法による行動解析実験を行った。社会実験の前後で撮影された動画（2022年8月27日（土）（ベンチ・カウンタ設置前）と10月22日（土）（ベンチ・カウンタ設置後））（図11）を用いて、実験を行った。動画の撮影は両日とも7:30～19:30まで行っている。1時間撮影し、1時間休憩する（12:30～14:30までは2時間休憩）、というサイクルを繰り返して、各6映像を撮影している。

##### Before

調査日: 8/27(土) 14:30-15:30  
温度 : 31~32°C



##### After

調査日: 10/22(土) 14:30-15:30  
温度 : 22~23°C



図11 社会実験前後の駅前

実験に用いたPCのスペックは以下の通りである。

CPU : Intel(R) Xeon(R) Silver 4110 CPU @ 2.10GHz

Memory : 64GB

SSD : 1TB

OS : Windows 10 22H2

GPU : GeForce RTX 3060 (12GB)

GPUは5～6万円程度のもを使用している。性能が良く、高価なGPUもあるが、姫路市などで業務に利用するのであれば、今回使ったGPUでも十分な性能である、と考えられる。

提案手法を適用し、画像処理をした結果を次の節から示す。



### 3-2. ヒートマップによる集中箇所検出

社会実験前後の動画からヒートマップを作成したものを示す(図12)。左図が社会実験において、ベンチやカウンタを設置する前の結果であり、右図が設置後の結果である。ベンチ・カウンタ設置後、そのエリアの利用が活用されていることがヒートマップからわかる。また、社会実験前後に関わらず、画像上部に赤い領域が左右に伸びており、この部分は日頃から人の行き来があることを示している。



(a) 14:30~15:30



(b) 16:30~17:30



(c) 16:30~17:30

図12 社会実験の前後におけるヒートマップの比較/右:社会実験前/左:社会実験後

### 3-3. 検出対象識別および行動追跡を用いた行動解析

行動識別および行動解析結果を図13に示す。提案手法において、識別および行動追跡ができることが確認できた。検出対象ごとに移動軌跡が確認できるようになるため、より詳細な情報を得ることができ、移動方向、移動速度、滞留時間などの計算に活用することができるため、行動解析への足掛かりになることがわかった。

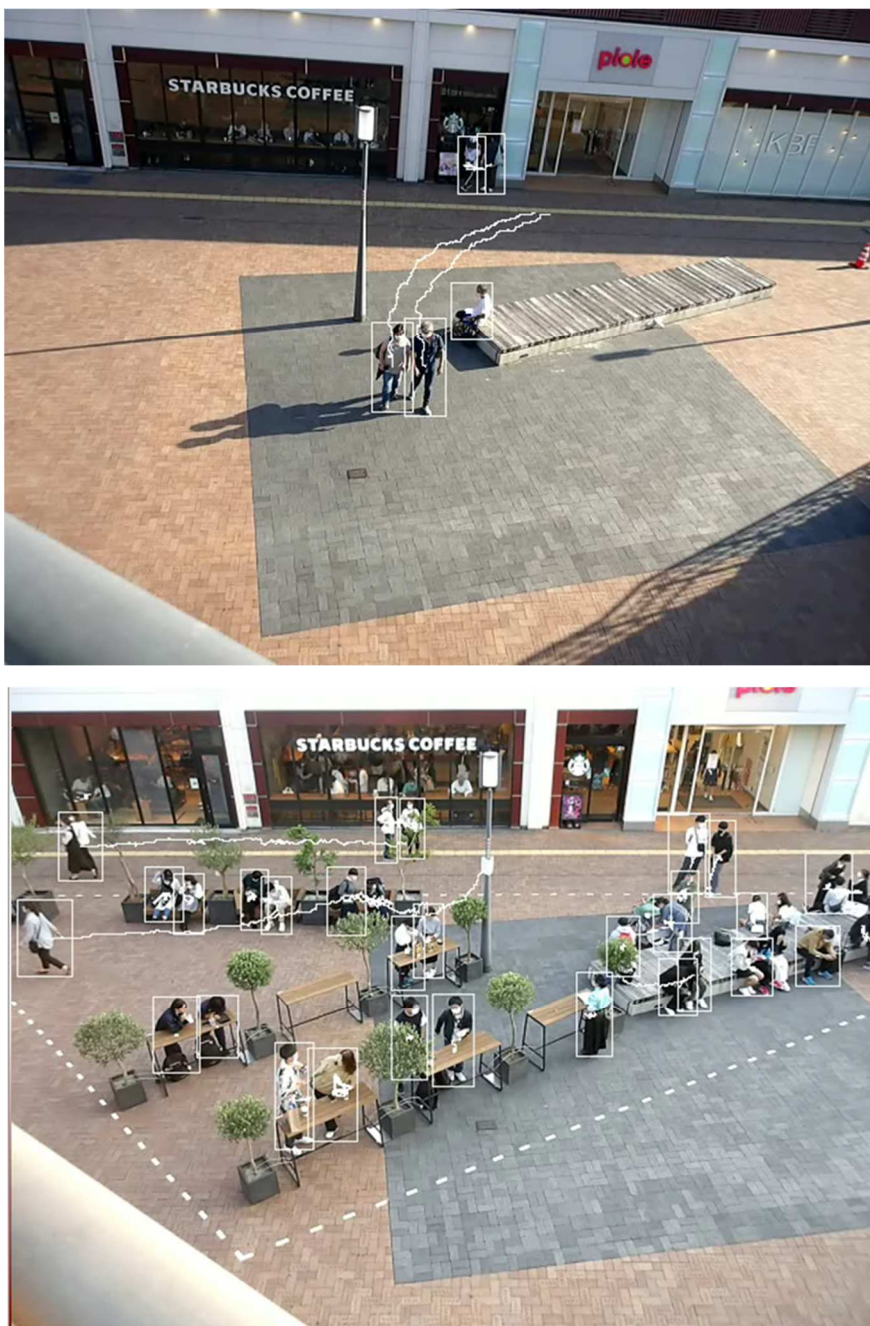


図13 社会実験の前後における行動追跡結果例／上：社会実験前／下：社会実験後  
検出枠（四角い白枠）と移動軌跡（白線）を描画している

### 3-4. 人流解析システム操作体験会

本研究の目的の1つに「市の職員が自ら解析することができ、業務に活用すること」を掲げている。そのため、連携課の職員に協力いただき、操作体験会を開催した。2024年2月9日（金）10時より、兵庫県立大学・姫路環境人間キャンパスにて、本研究に開発した行動解析支援システムシステムの操作体験会を実施した。姫路環境人間キャンパスA棟3階A303室（画像メディア研究室）にて、連携課である姫路市・都市計画課・有方聡氏、デジタル戦略室・田中英太郎氏に操作体験をしていただいた。有方氏は大学時代、プログラミングの講義を受講した経験があり、今回の体験会について、不安は少なかったようである。田中氏は学生にプログラミングのようなコンピュータでの作業の経験はなく、体験前は多少不安があると口にしていた。

**解析システム操作体験**

兵庫県立大学 環境人間学部 木村敏文  
(画像メディア研究室)

日時： 2月9日（金）10時より  
場所： 兵庫県立大学 姫路環境人間キャンパス A棟3階 A302室

実施内容：  
令和5年度姫路市大学発まちづくり研究助成事業における研究課題「動画を用いた機械学習による行動解析」において、開発しているシステムの操作体験を実施します。姫路市職員が活用できるようにする、という目標に対して、姫路市職員の方々にシステムの操作体験や評価していただきたいと思っております。

体験いただく内容は、以下のとおりです。

- 1) 撮影動画の扱い方について
- 2) 機械学習を用いた人検出による人数カウント（図1）
- 3) 開発ソフトウェアを用いた滞留解析（図2）

慣れない操作で難しい部分もあると思いますが、できるだけ簡単に操作できるように準備しておきますので、ご参加下さい。

以上





図1 機械学習を用いた人検出結果例図2 滞留解析例

図14 操作体験会実施案内資料

操作体験会では、あらかじめ、YOLOv7 をインストールした状態の PC を 2 台準備した。市業務で活用する前に行動解析支援システムの動作環境構築が必要となるが、体験会では時間の関係上省いた。また、事前に連携課の方々と打ち合わせをした際、ヒートマップとして、集中箇所がわかるだけでも役立つ、という意見をあったため、ヒートマップの作成を体験してもらうこととした。体験会では以下のスライドを基に説明を行った。

### 今日の実施内容

- 令和5年度姫路市大学発まちづくり研究助成事業における研究課題「動画を用いた機械学習による行動解析」において、開発しているシステムの操作体験を実施します。
- 姫路市職員が活用できるようにする、という目標に対して、開発システムの操作体験や評価していただきたいと思います。

### 今日の操作体験（ヒートマップを作ってみる）

- 1)撮影動画の扱い方について
- 2)機械学習を用いた人検出による人数カウント
- 3)開発ソフトウェアを用いた滞留解析




### 主に使うコマンドは 2 つ

- `python detect.py --source ファイル名 --weights yolov7-w6-person.pt --conf 0.4 --save-txt --save-conf`  
(機械学習 (yolov7) を用いた人検出)
- `python h20240209.py`  
(開発プログラムによるヒートマップ)

### 本当は……

- yolov7をインストールする必要がある
- Windowsで yolov7 をインストールするには…  
これをやると1日では済まなくなるかもしれないので（省略）

↓

インストールしているものとして、話しを進めます

### 1)撮影動画の扱い方について

- yolov7 は多様な動画形式を扱うことができる
- それでもできないものもある
- そんなときはどうするか？ => **ffmpeg (フリーソフト)** を使う
- **ffmpeg** は動画形式の変換が可能
- 使える動画形式に変換することで、yolov7で扱うことができるようになる
- こんなコマンドで使える（今日は省略）  
=> `ffmpeg -i 旧ファイル名.mov 新ファイル名.mp4`

### 2)機械学習を用いた人検出

- yolov7 を用いた人検出
- 予め学習したモデルを使えば様々なものが検出できる
- 今回使うのは人の検出に特化したモデル (yolov7-w6-person.pt)
- 使い方は…
  - yolov7フォルダに動画ファイルをコピー
  - `conda activate yolov7`
  - `cd yolov7`
  - `python detect.py --source 動画ファイル名 --weights yolov7-w6-person.pt --conf 0.4 --save-txt --save-conf`
- あとは自動で人を検出してくれる

### 3)開発ソフトウェアを用いた滞留解析

- 開発ソフトウェアを用いてヒートマップを作成
- ヒートマップから滞留解析の可能性を検討
- コマンドとしては…これだけ
  - `cd heatmap`
  - `python h20240209.py`
  - 動画ファイルを指定
  - 最初の検出データファイルを指定
  - ヒートマップ（動画、最終シーン）

### 3)開発ソフトウェアを用いた滞留解析

- ヒートマップ




図 15 操作体験会で提示した説明スライド

提案システムはコンピュータを普段使っている方でもほとんど使うことのないコマンドプロンプトを利用する。難しそうに感じる方も多いとは思いますが、慣れると簡単と感じてくれる方もいる。今回、動画を用意すれば、主なコマンドは2つだけでヒートマップが作成でき、人が集中していく時間的変化を確認することができるようになる。この体験を姫路市職員2名の方に体験していただいた。



図 16 操作体験会の様子

体験いただいた職員の方々の感想としては、

- ・ コマンドを打つと聞いて、敷居が高そうだったが、やってみると難しくはなかった
- ・ 意外と簡単にヒートマップが作れて、びっくりした
- ・ ヒートマップが作れるだけでも役立つことはありそう

ということでした。できるだけ簡単に作業ができるように、手法とプログラムで工夫をしているので、その点が評価された、と感じている。また、この体験会で2台のPCには違うタイプのGPUをセットしていた。1台はシステムの開発に用いた GeForce RTX3060 - 12GB、もう1台は GeForce GTX1650 - 4GB である。GTX1650 は価格も比較的安く、お手頃なGPUである。電源消費量も少ないため、お試しで使うには良いGPUである、と感じている。体験いただいた職員の方々に2台のPCを使っていただきましたが、同じ動画(5分くらい)から人を検出するのに、RTX3060 搭載 PC では5分30秒くらいで終わったのに対して、GTX1650 搭載 PC では50分以上かかる結果となった。提案手法の処理はラップトップPCでも処理可能だが、性能的には多くの時間を必要となる。GPUにより高速化が可能となる、と考えていたが、GTX1650 では思っているほどの性能はない、ということがわかった。この手法の上では、実作業に活用することを考えるとそれなり良いGPUが必須であることがわかった。普段、市役所での各部署で利用されているPCで提案手法がそのまま利用することが望ましいが、現状ではそれなりに良いGPUを搭載したPCが必要になり、どのように市の業務に活用していくのか、を考える必要がある。

機器に対する問題点を再認識できた体験会において、市職員2名の方に提案システムの操作体験をしていただき、市の職員の方も操作が十分可能である、という感触を得ることができたことは大きな収穫であった。

## 第4章 まとめ

本研究では、動画から把握できる基本的な情報(人数、人の集中箇所等)を取得するため、機械学習を用いた解析手法を提案した。YOLO を用いた人検出を組み込み、検出枠と位置情報を得て、この情報を活用したヒートマップ作成方法や検出対象の識別、行動追跡を手法とプログラムを開発した。また、本研究で大きな目標であった「市の職員が自ら解析することができ、業務に活用すること」については、市の職員の方に開発プログラムを操作いただき、基本的なデータを取得することが可能であるのか、を検討した。実際に連携課の職員の方々に操作体験をしていただくことでその可能性を見出すことができた。また、本研究での成果は、機械学習を用いて、動画から 1) ヒートマップを作成、2) 検出対象の識別、3) 検出対象の解析、をできるようにしたことである。

今後の課題と展望は以下のことが挙げられる

1つめとして、姫路市の各部署にどのように普及させていくのか、である。実行環境構築については、概要版、詳細版の導入マニュアルを文末の付録に示す。導入にはコンピュータの知識が多少なりとも必要なため、ヒートマップを作成するためのコマンド利用に比べて、難易度が高いと考えられる。導入を希望される場合、引き続き、支援していく必要がある、と考えている。

2つめとして、検出対象の識別の向上、および、解析方法の検討である。本研究ではイベント時や日常における人流調査の基本データとなる部分を機械学習、AI 技術を用いて、軽減するための手法を提案した。このことで一定の成果は得られるが、業務に必要なデータは多種多様であるため、カバーできていない部分も多々あると考える。この研究をきっかけにこんなことができないか、という意見があれば、できる、できない、も含めた議論ができる。本研究の成果はすべての業務に役立つものではない。業務ごとに違う手法が必要であるため、多くの職員の方に興味を持ってもらい、希望を聞くことで、次のステップの研究に進めることが期待でき、業務の簡略化や仕事量の軽減につながる可能性がある、と考えている。

3つめとして、リアルタイム処理の可能性を探ることである。本研究では撮影後に研究室で映像を処理している。ラップトップ PC では性能不足の面があるため、研究室にある高性能デスクトップ PC を使った処理が必要である。リアルタイム処理は今回提案した手法と共に撮影も同時にするため、さらに高性能な PC が必要となる。また、リアルタイム処理を行うためには無駄な処理を省くため、用途に合った処理が必要である。どのような場所でどのような処理を必要としているのか、調査する必要もある、と考える。この課題を実施するためには撮影も含めた新たな手法を提案することと共に、高性能な PC の調達も必要である、と考えるため、今後の課題とする。

## 参考文献

1. 国土交通省都市局, スマート・プランニング実践の手引き～個人単位の行動データに基づく新たなまちづくり～【第二版】, 2018
2. 多田功, 加古川市におけるスマートシティの取り組みについて, 区画整理 65(10), pp.29-37, 2022
3. J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You Only Look Once: Unified, Real-Time Object Detection, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 779-788, 2016
4. Toshifumi Kimura, Mizue Ohashi, Karl Crailsheim, Thomas Schmickl, Ryuichi Okada, Gerald Radspieler, Hidetoshi Ikeno, Development of a new method to track multiple honey bees with complex behaviors on a flat laboratory arena, PLoS ONE 9(1): e84656, 2014
5. Toshifumi Kimura, Mizue Ohashi, Karl Crailsheim, Thomas Schmickl, Ryuichi Okada, Gerald Radspieler, Tejiro Isokawa, Hidetoshi Ikeno, A Heuristic Trajectory Decision Method to Enhance the Tracking Performance of Multiple Honeybees on a Flat Laboratory Arena, Transactions of the Institute of Systems, Control and Information Engineers, 32(3), 113-122, 2019
6. Chien-Yao Wang, Alexey Bochkovskiy, Hong-Yuan Mark Liao, YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 7464-7475, 2023

## 謝辞

本研究を実施するにあたり、姫路市都市局まちづくり部都市計画課の皆様、デジタル戦略本部デジタル戦略室の皆様にご感謝申し上げます。また、現状の調査インタビューに協力いただきました加古川市・企画部 政策企画課の皆様にご感謝申し上げます。兵庫県立大学・環境人間学部・太田尚孝教授、大学院工学研究科・磯川悌次郎准教授、兵庫県立大学・環境人間学部・画像メディア研究室・学生の皆さんにも研究の実施にご協力いただきました。ここに記してお礼申し上げます。

今後、本研究で開発したシステム（手順、プログラム）が姫路市の職員の皆さんに活用いただき、姫路市の行政サービス向上につながるツールになれば幸いです。

2024年3月

研究代表者： 兵庫県立大学環境人間学部 講師 木村敏文



## 付録

### 行動解析支援システム・環境構築ガイド (簡易版)



## 行動解析支援システム・環境構築ガイド（簡易版）

行動解析支援システムを利用するためには多少、性能の良いPCを必要としている。以下に最小限のPCスペックと考えられるものを示す。

【PCスペック】（2024年2月現在）

- ・CPU：intel Core i5 以上
- ・メモリ：16GB 以上
- ・SSD：500GB 以上
- ・OS：Windows 11（Windows 10でも動作は確認）
- ・GPU：nvidia GeForce RTX 2060 以上
- ・電源：GPUに合った電源容量を選ぶこと（最低でも500W以上）

## 行動解析支援システム・環境構築手順

### 1. anaconda インストール

<https://www.anaconda.com/download>

（それなりに時間がかかる⇒約20～30分）

### 2. YOLOv7 動作環境構築

アプリから anaconda prompt を起動

コマンドを実行

```
conda create -n yolov7 python=3.9
```

YOLOv7 動作環境への移行

```
conda activate yolov7
```

### 3. (GPUがある場合のみ) CUDA インストール

CUDA 11.8 をダウンロード (yolov7 は CUDA 11.8 を使うこと)

<https://developer.nvidia.com/cuda-11-8-0-download-archive>

[Windows]->[x86\_64]->[11]->[exe(local)]を選択

Windows のバージョンによって、[11]->[10]に変更

CUDA をインストール（ファイルサイズ：3GB程度）

### 4. PyTorch インストール

PyTorch のホームページで conda 用インストールコマンドを確認する

<https://pytorch.org/get-started/locally/>

コマンドを実行

```
conda activate yolov7
```

```
python -m pip install --upgrade pip
```

PyTorch のインストール

(GPU があるとき) / 20240227 HP で確認したコマンド (1 行で入力して下さい)

コマンドを実行

```
conda install pytorch torchvision torchaudio  
pytorch-cuda=11.8 -c pytorch -c nvidia
```

(GPU がないとき) / 20240227 HP で確認したコマンド

```
conda install pytorch torchvision torchaudio cpuonly -c pytorch
```

## 5. Git インストール

Git 64bit Windows (Git-2.44.0-64-bit.exe) をダウンロード

<https://git-scm.com/download/win>

Git をインストール (Git-2.44.0-64-bit.exe をダブルクリック)

## 6. YOLOv7 インストール

アプリから anaconda prompt を起動

コマンドを実行 (YOLOv7 の動作環境へ移行)

```
conda activate yolov7
```

YOLOv7 インストール

```
git clone https://github.com/WongKinYiu/yolov7.git
```

YOLOv7 実行フォルダの移動

```
cd yolov7
```

YOLOv7 動作環境の復元

```
pip install -r requirements.txt
```

YOLOv7 学習モデルのダウンロード

<https://github.com/WongKinYiu/yolov7/releases/tag/v0.1>

yolov7.pt をダウンロード

yolov7-w6-person.pt をダウンロード

ダウンロードした 2 つのファイルを yolov7 フォルダに移動

## 7. YOLOv7 検出テスト

YOLOv7 の検出テストを行う

(コマンドを実行 / 1 行で入力)

```
python detect.py --weights yolov7.pt  
--source inference/images/horses.jpg
```

(runs/detect フォルダの exp フォルダに結果がある)

(検出するたびに exp のあとに連番で数字が付加される)

(馬が検出できていれば、インストール完了)

※行動解析支援システムの利用環境が整いました※



## 付録

### 行動解析支援システム・環境構築ガイド (詳細版)



## 行動解析支援システム・環境構築ガイド（詳細版）

行動解析支援システムを利用するためには多少、性能の良いPCを必要としている。以下に最小限のPCスペックと考えられるものを示す。

【PCスペック】（2024年2月現在）

- ・CPU：intel Core i5 以上
- ・メモリ：16GB 以上
- ・SSD：500GB 以上
- ・OS：Windows 11（Windows 10でも動作は確認）
- ・GPU：nvidia GeForce RTX 2060 以上
- ・電源：GPUに合った電源容量を選ぶこと（最低でも500W以上）

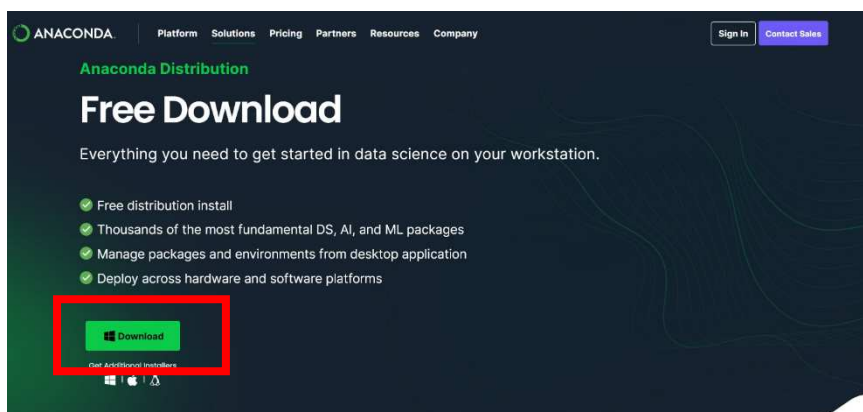
## 行動解析支援システム・環境構築手順

### 1. anaconda インストール

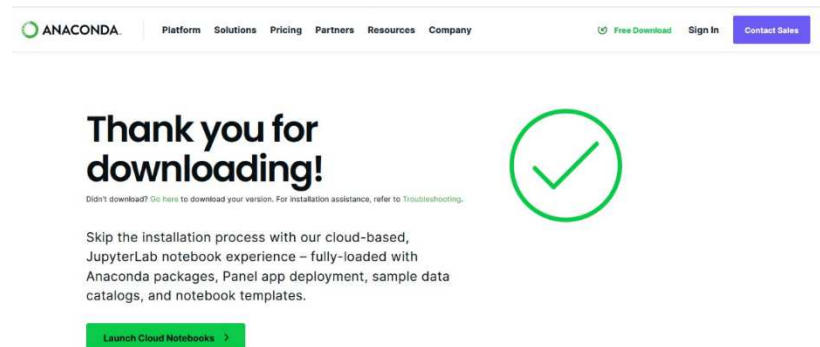
<https://www.anaconda.com/download>

（それなりに時間がかかる⇒約20～30分）

#### 【ダウンロード画面】



#### 【ダウンロード完了画面】





## 【インストール画面】

The image displays a sequence of eight screenshots from the Anaconda3 2023.09-0 (64-bit) Setup wizard, arranged in a 4x2 grid. Each screenshot shows a different step in the installation process, with the Anaconda logo and title bar visible at the top of each window.

- Top-Left:** "Welcome to Anaconda3 2023.09-0 (64-bit) Setup". The window contains the Anaconda logo, a welcome message, and instructions to click "Next" to continue.
- Top-Right:** "License Agreement". It displays the End User License Agreement (EULA) for Anaconda Distribution, with a scrollable text area and "I Agree" and "Cancel" buttons.
- Middle-Left:** "Select Installation Type". It offers two options: "Just Me (recommended)" (selected) and "All Users (requires admin privileges)".
- Middle-Right:** "Choose Install Location". It prompts the user to choose a folder for installation, showing a text field with "C:\Users\kimura\anaconda3" and a "Browse..." button. It also indicates "Space required: 5.7 GB" and "Space available: 137.8 GB".
- Bottom-Left:** "Advanced Installation Options". It lists several options with checkboxes: "Create start menu shortcuts", "Add Anaconda3 to my PATH environment variable", "Register Anaconda3 as my default Python 3.11", and "Clear the package cache upon completion".
- Bottom-Right:** "Installation Complete". A green progress bar indicates 100% completion, and a "Show details" button is present.
- Bottom-Left (Final):** "Completing Anaconda3 2023.09-0 (64-bit) Setup". It thanks the user and provides helpful resources, with checkboxes for "Launch Anaconda Navigator" and "Getting Started with Anaconda Distribution".
- Bottom-Right (Final):** "Anaconda3 2023.09-0 (64-bit) Code with Anaconda in the Cloud". It promotes cloud-based development with Anaconda Notebooks and features the Anaconda and Jupyter logos.

# Welcome to Anaconda!

Here are some useful resources to help you get started.



Create your free [Anaconda Cloud](#) account today to get access to training materials, how-to videos, and expert insights, all free for a limited time to Nucleus members.

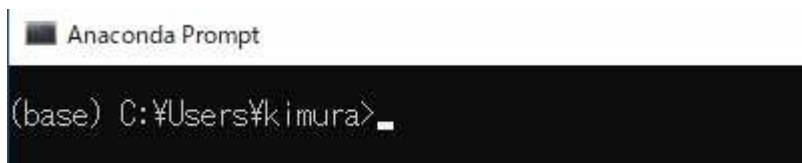
[Register for Free](#) >

## 2. YOLOv7 動作環境構築

アプリから anaconda prompt を起動



Anaconda Prompt が起動する



コマンドを実行

```
conda create -n yolov7 python=3.9
```

```
(base) C:\Users\kimura>conda create -n yolov7 python=3.9
Collecting package metadata (current_repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 23.7.4
  latest version: 24.1.2

Please update conda by running

  $ conda update -n base -c defaults conda

Or to minimize the number of packages updated during conda update use

  conda install conda=24.1.2

## Package Plan ##
```

y を入力し、enter キーを押す

```
Anaconda Prompt - conda create -n yolov7 python=3.9

The following packages will be downloaded:

-----
package                                     build                                     size
-----
ca-certificates-2023.12.12                 haa95532_0                             127 KB
openssl-3.0.13                              h2bfff1b_0                             7.4 MB
pip-23.3.1                                  py39haa95532_0                         2.8 MB
python-3.9.18                               h1aa4202_0                             19.4 MB
setuptools-68.2.2                          py39haa95532_0                         933 KB
tzdata-2024a                               h04d1e81_0                             116 KB
wheel-0.41.2                                py39haa95532_0                         126 KB
-----
Total:                                     30.9 MB

The following NEW packages will be INSTALLED:

ca-certificates pkgs/main/win-64::ca-certificates-2023.12.12-haa95532_0
openssl         pkgs/main/win-64::openssl-3.0.13-h2bfff1b_0
pip             pkgs/main/win-64::pip-23.3.1-py39haa95532_0
python         pkgs/main/win-64::python-3.9.18-h1aa4202_0
setuptools     pkgs/main/win-64::setuptools-68.2.2-py39haa95532_0
sqlite        pkgs/main/win-64::sqlite-3.41.2-h2bfff1b_0
tzdata        pkgs/main/noarch::tzdata-2024a-h04d1e81_0
vc            pkgs/main/win-64::vc-14.2-h21ff451_1
vs2015_runtime pkgs/main/win-64::vs2015_runtime-14.27.29016-h5e58377_2
wheel         pkgs/main/win-64::wheel-0.41.2-py39haa95532_0

Proceed ([y]/n)? y_
```

YOLOv7 の動作環境構築が完了

```
Anaconda Prompt

openssl         pkgs/main/win-64::openssl-3.0.13-h2bfff1b_0
pip             pkgs/main/win-64::pip-23.3.1-py39haa95532_0
python         pkgs/main/win-64::python-3.9.18-h1aa4202_0
setuptools     pkgs/main/win-64::setuptools-68.2.2-py39haa95532_0
sqlite        pkgs/main/win-64::sqlite-3.41.2-h2bfff1b_0
tzdata        pkgs/main/noarch::tzdata-2024a-h04d1e81_0
vc            pkgs/main/win-64::vc-14.2-h21ff451_1
vs2015_runtime pkgs/main/win-64::vs2015_runtime-14.27.29016-h5e58377_2
wheel         pkgs/main/win-64::wheel-0.41.2-py39haa95532_0

Proceed ([y]/n)? y

Downloading and Extracting Packages
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

##
## To activate this environment, use
##
##   $ conda activate yolov7
##
## To deactivate an active environment, use
##
##   $ conda deactivate

(base) C:\Users\kimura>
```

## YOLOv7 動作環境への移行

**conda activate yolov7**

```
Anaconda Prompt

Proceed ([y]/n)? y

Downloading and Extracting Packages

Preparing transaction: done
Verifying transaction: done
Executing transaction: done

#
# To activate this environment, use
#
#   $ conda activate yolov7
#
# To deactivate an active environment, use
#
#   $ conda deactivate

(base) C:\Users\kimura>conda activate yolov7
```

## 先頭の文字列が (yolov7) に変わる

```
Anaconda Prompt

Proceed ([y]/n)? y

Downloading and Extracting Packages

Preparing transaction: done
Verifying transaction: done
Executing transaction: done

#
# To activate this environment, use
#
#   $ conda activate yolov7
#
# To deactivate an active environment, use
#
#   $ conda deactivate

(base) C:\Users\kimura>conda activate yolov7

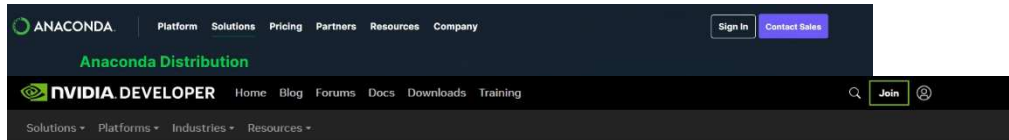
(yolov7) C:\Users\kimura>
```

### 3. (GPUがある場合のみ) CUDA インストール

CUDA 11.8 をダウンロード (yolov7 は CUDA 11.8 を使うこと)

<https://developer.nvidia.com/cuda-11-8-0-download-archive>

【ダウンロード画面】



## CUDA Toolkit 11.8 Downloads

### Select Target Platform

Click on the green buttons that describe your target platform. Only supported platforms will be shown. By downloading and using the software, you agree to fully comply with the terms and conditions of the [CUDA EULA](#).

<b>Operating System</b>	<input checked="" type="button" value="Linux"/>	<input checked="" type="button" value="Windows"/>
-------------------------	---	---

<b>Resources</b>	
<ul style="list-style-type: none"><li>• <a href="#">CUDA Documentation/Release Notes</a></li><li>• <a href="#">MacOS Tools</a></li><li>• <a href="#">Training</a></li><li>• <a href="#">Sample Code</a></li><li>• <a href="#">Forums</a></li></ul>	<ul style="list-style-type: none"><li>• <a href="#">Archive of Previous CUDA Releases</a></li><li>• <a href="#">FAQ</a></li><li>• <a href="#">Open Source Packages</a></li><li>• <a href="#">Submit a Bug</a></li><li>• <a href="#">Tarball and Zip Archive Deliverables</a></li></ul>

[Windows]->[x86\_64]->[11]->[exe(local)] を選択

Windows のバージョンによって、[11]->[10]に変更する

## CUDA Toolkit 11.8 Downloads

### Select Target Platform

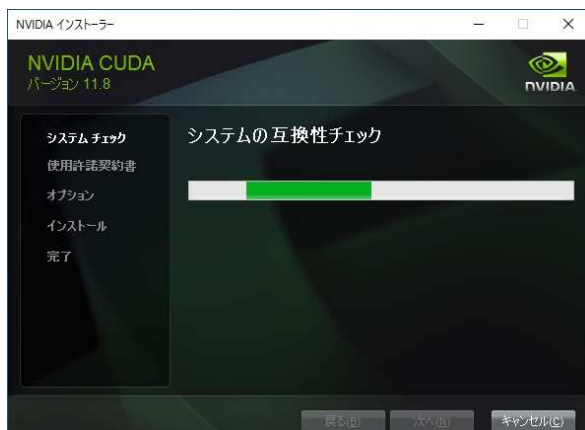
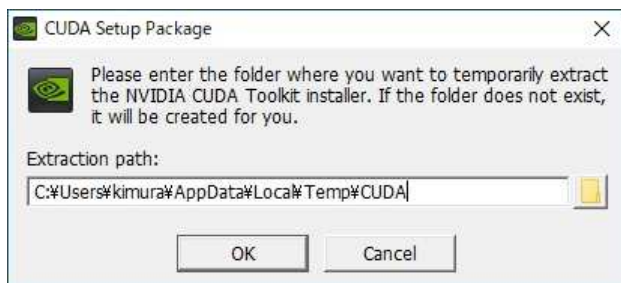
Click on the green buttons that describe your target platform. Only supported platforms will be shown. By downloading and using the software, you agree to fully comply with the terms and conditions of the [CUDA EULA](#).

<b>Operating System</b>	<input checked="" type="button" value="Linux"/>	<input checked="" type="button" value="Windows"/>			
<b>Architecture</b>	<input checked="" type="button" value="x86_64"/>				
<b>Version</b>	<input checked="" type="button" value="10"/>	<input checked="" type="button" value="11"/>	<input type="button" value="Server 2016"/>	<input type="button" value="Server 2019"/>	<input type="button" value="Server 2022"/>
<b>Installer Type</b>	<input checked="" type="button" value="exe (local)"/>		<input type="button" value="exe (network)"/>		

<b>Download Installer for Windows 11 x86_64</b>	
The base installer is available for download below.	
> Base Installer	<input type="button" value="Download (3.0 GB)"/>

## CUDA をインストール (ファイルサイズ : 3 GB 程度)



これで、CUDA のインストール完了

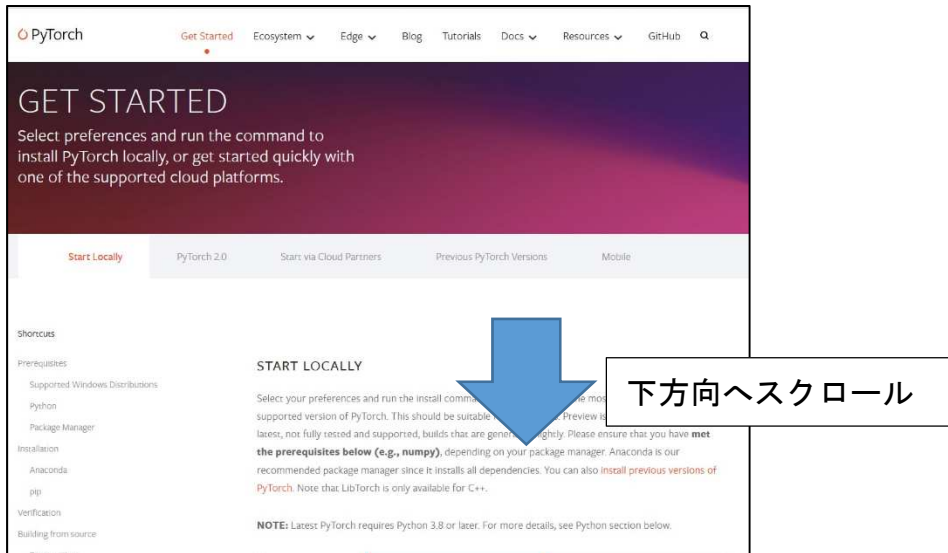
A

## 4. PyTorch のインストール

PyTorch のホームページで conda 用インストールコマンドを確認する

<https://pytorch.org/get-started/locally/>

【インストールコマンド確認画面】



(GPUがあるとき) / 20240227 HPで確認したコマンド

```
conda install pytorch torchvision torchaudio  
pytorch-cuda=11.8 -c pytorch -c nvidia
```

### START LOCALLY

Select your preferences and run the install command. Stable represents the most currently tested and supported version of PyTorch. This should be suitable for many users. Preview is available if you want the latest, not fully tested and supported, builds that are generated nightly. Please ensure that you have met the prerequisites below (e.g., numpy), depending on your package manager. Anaconda is our recommended package manager since it installs all dependencies. You can also install previous versions of PyTorch. Note that LibTorch is only available for C++.

**NOTE:** Latest PyTorch requires Python 3.8 or later. For more details, see Python section below.

PyTorch Build	Stable (2.2.1)	Preview (Nightly)		
Your OS	Linux	Mac	Windows	
Package	Conda	Pip	LibTorch	Source
Language	Python	C++ / Java		
Compute Platform	CUDA 11.8	CUDA 12.1	ROCm 5.7	CPU
Run this Command:	<pre>conda install pytorch torchvision torchaudio pytorch-cuda=11.8 -c pytorch -c nvidia</pre>			

## (GPUがないとき) / 20240227 HPで確認したコマンド

```
conda install pytorch torchvision torchaudio cpuonly -c pytorch
```

START LOCALLY

Select your preferences and run the install command. Stable represents the most currently tested and supported version of PyTorch. This should be suitable for many users. Preview is available if you want the latest, not fully tested and supported, builds that are generated nightly. Please ensure that you have **met the prerequisites below (e.g., numpy)**, depending on your package manager. Anaconda is our recommended package manager since it installs all dependencies. You can also [install previous versions of PyTorch](#). Note that LibTorch is only available for C++.

**NOTE:** Latest PyTorch requires Python 3.8 or later. For more details, see Python section below.

PyTorch Build	Stable (2.2.1)	Preview (Nightly)		
Your OS	Linux	Mac	Windows	
Package	Conda	Pip	LibTorch	Source
Language	Python	C++ / Java		
Compute Platform	CUDA 11.8	CUDA 12.1	ROCm 5.7	CPU
Run this Command:	conda install pytorch torchvision torchaudio cpuonly -c pytorch			

## PyTorch のインストール

コマンドを実行

```
conda activate yolov7
python -m pip install --upgrade pip
```

```
Anaconda Prompt
(base) C:\Users\kimura>conda activate yolov7
(yolov7) C:\Users\kimura>conda install pytorch torchvision torchaudio pytorch-cuda=11.8 -c pytorch -c nvidia
```

## (GPUがあるとき) / 1行で入力して下さい

コマンドを実行

```
conda install pytorch torchvision torchaudio
pytorch-cuda=11.8 -c pytorch -c nvidia
```



y を入力し、enter キーを押す

```
Anaconda Prompt - conda install pytorch torchvision torchaudio pytorch-cuda=11.8 -c pytorch -c nvidia
```

```
cuda-cccl          nvidia/win-64::cuda-cccl-12.3.101-0
cuda-cudart        nvidia/win-64::cuda-cudart-11.8.89-0
cuda-cudart-dev    nvidia/win-64::cuda-cudart-dev-11.8.89-0
cuda-cupti         nvidia/win-64::cuda-cupti-11.8.87-0
cuda-libraries     nvidia/win-64::cuda-libraries-11.8.0-0
cuda-libraries-dev nvidia/win-64::cuda-libraries-dev-11.8.0-0
cuda-nvrtc         nvidia/win-64::cuda-nvrtc-11.8.89-0
cuda-nvrtc-dev     nvidia/win-64::cuda-nvrtc-dev-11.8.89-0
cuda-nvtx          nvidia/win-64::cuda-nvtx-11.8.86-0
cuda-profiler-api  nvidia/win-64::cuda-profiler-api-12.3.101-0
cuda-runtime       nvidia/win-64::cuda-runtime-11.8.0-0
libcublas          nvidia/win-64::libcublas-11.11.3.6-0
libcublas-dev      nvidia/win-64::libcublas-dev-11.11.3.6-0
libcufft           nvidia/win-64::libcufft-10.9.0.58-0
libcufft-dev       nvidia/win-64::libcufft-dev-10.9.0.58-0
libcurand          nvidia/win-64::libcurand-10.3.4.107-0
libcurand-dev      nvidia/win-64::libcurand-dev-10.3.4.107-0
libcusolver        nvidia/win-64::libcusolver-11.4.1.48-0
libcusolver-dev    nvidia/win-64::libcusolver-dev-11.4.1.48-0
libcuspars         nvidia/win-64::libcuspars-11.7.5.86-0
libcuspars-dev     nvidia/win-64::libcuspars-dev-11.7.5.86-0
libnpp             nvidia/win-64::libnpp-11.8.0.86-0
libnpp-dev         nvidia/win-64::libnpp-dev-11.8.0.86-0
libnvjpeg          nvidia/win-64::libnvjpeg-11.9.0.86-0
libnvjpeg-dev      nvidia/win-64::libnvjpeg-dev-11.9.0.86-0
pytorch-cuda       pytorch/win-64::pytorch-cuda-11.8-h24eeafa_5
```

```
Proceed ([y]/n)? y
```

これが表示すれば、PyTorch（GPU あり版）のインストールは完了

```
Anaconda Prompt - conda install pytorch torchvision torchaudio pytorch-cuda=11.8 -c pytorch -c nvidia
```

```
Proceed ([y]/n)? y
```

```
Downloading and Extracting Packages
```

```
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
```

```
(yolov7) C:\Users\kimura>
```

(GPUがないとき)

```
conda install pytorch torchvision torchaudio cpuonly -c pytorch
```

```
Anaconda Prompt
(base) C:\Users\kimura>conda activate yolov7
(yolov7) C:\Users\kimura>python -m pip install --upgrade pip
Requirement already satisfied: pip in c:\users\kimura\anaconda3\envs\yolov7\lib\site-packages (23.3.1)
Collecting pip
  Downloading pip-24.0-py3-none-any.whl.metadata (3.6 kB)
  Downloading pip-24.0-py3-none-any.whl (2.1 MB)
----- 2.1/2.1 MB 7.1 MB/s eta 0:00:00
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 23.3.1
    Uninstalling pip-23.3.1:
      Successfully uninstalled pip-23.3.1
  Successfully installed pip-24.0
(yolov7) C:\Users\kimura>conda install pytorch torchvision torchaudio cpuonly -c pytorch
```

yを入力し、enter キーを押す

```
Anaconda Prompt - conda install pytorch torchvision torchaudio cpuonly -c pytorch
mkl_fft          pkgs/main/win-64::mkl_fft-1.3.8-py39h2bfff1b_0
mkl_random       pkgs/main/win-64::mkl_random-1.2.4-py39h59b6b97_0
mpc              pkgs/main/win-64::mpc-1.1.0-h7edee0f_1
mpfr            pkgs/main/win-64::mpfr-4.0.2-h62dcd97_1
mpir            pkgs/main/win-64::mpir-3.0.0-hec2e145_1
mpmath          pkgs/main/win-64::mpmath-1.3.0-py39haa95532_0
networkx        pkgs/main/win-64::networkx-3.1-py39haa95532_0
numpy           pkgs/main/win-64::numpy-1.26.4-py39h055cbcc_0
numpy-base     pkgs/main/win-64::numpy-base-1.26.4-py39h65a83cf_0
openjpeg        pkgs/main/win-64::openjpeg-2.4.0-h4fc8c34_0
pillow          pkgs/main/win-64::pillow-10.2.0-py39h2bfff1b_0
pysocks         pkgs/main/win-64::pysocks-1.7.1-py39haa95532_0
pytorch         pytorch/win-64::pytorch-2.2.1-py3.9_cpu_0
pytorch-mutex  pytorch/noarch::pytorch-mutex-1.0-cpu
pyyaml          pkgs/main/win-64::pyyaml-6.0.1-py39h2bfff1b_0
requests        pkgs/main/win-64::requests-2.31.0-py39haa95532_1
sympy           pkgs/main/win-64::sympy-1.12-py39haa95532_0
tbb             pkgs/main/win-64::tbb-2021.8.0-h59b6b97_0
torchaudio     pytorch/win-64::torchaudio-2.2.1-py39_cpu
torchvision     pytorch/win-64::torchvision-0.17.1-py39_cpu
typing_extensions pkgs/main/win-64::typing_extensions-4.9.0-py39haa95532_1
urllib3         pkgs/main/win-64::urllib3-2.1.0-py39haa95532_1
win_inet_pton  pkgs/main/win-64::win_inet_pton-1.1.0-py39haa95532_0
xz             pkgs/main/win-64::xz-5.4.5-h8cc25b3_0
yaml           pkgs/main/win-64::yaml-0.2.5-he774522_0
zlib           pkgs/main/win-64::zlib-1.2.13-h8cc25b3_0
zstd           pkgs/main/win-64::zstd-1.5.5-hd43e919_0

Proceed ([y]/n)? y
```

しばらく待てば、PyTorch (GPUなし版) のインストールは完了

## 5. Git インスト

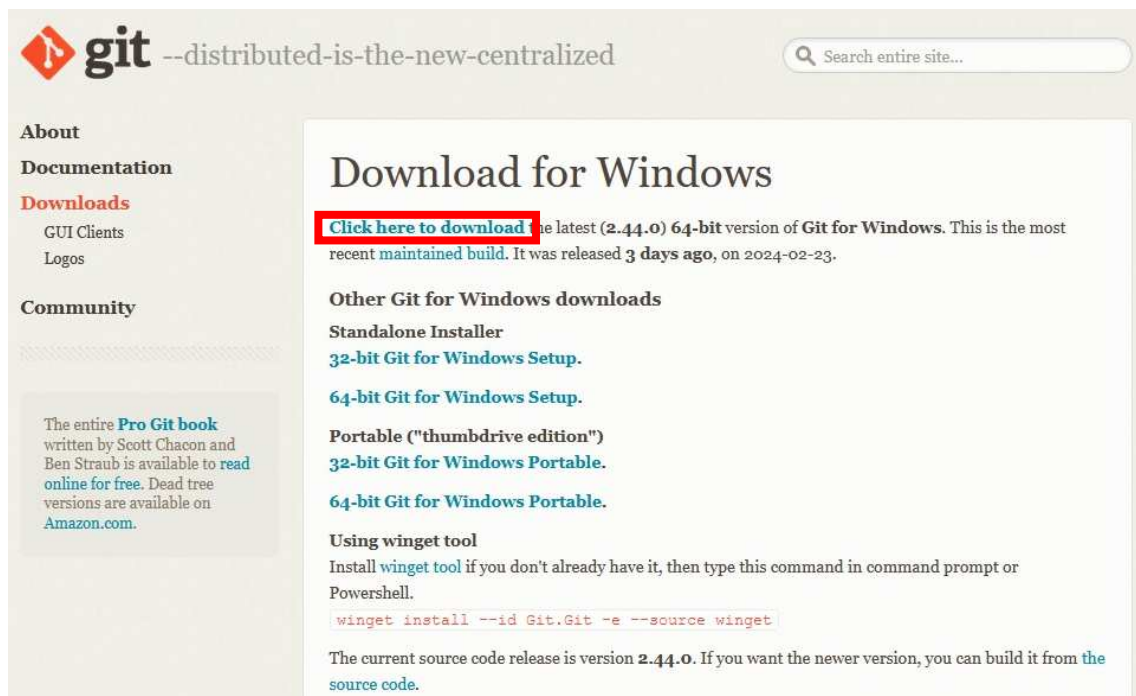
Git 64bit Windows (Git-2.44.0-64-bit.exe) をダウンロード  
<https://git-scm.com/download/win>

【ダウンロード画面】

windows を選択し、クリックする

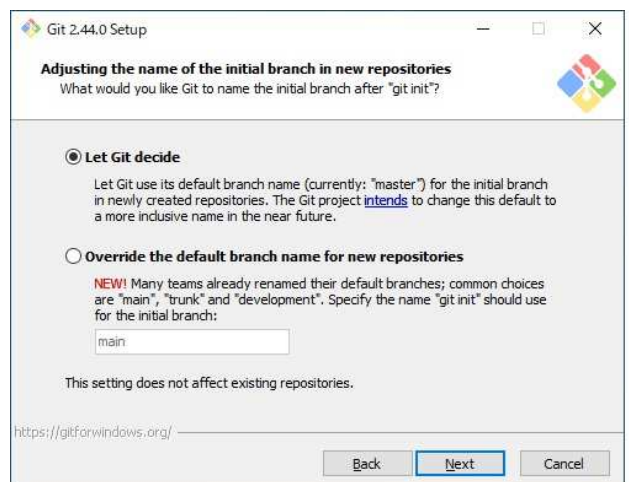
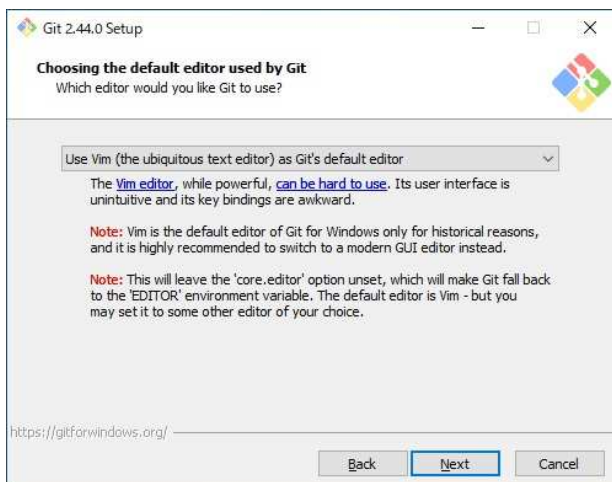
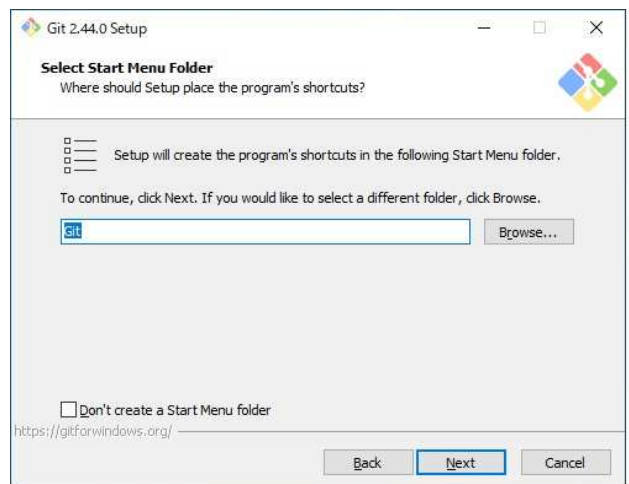
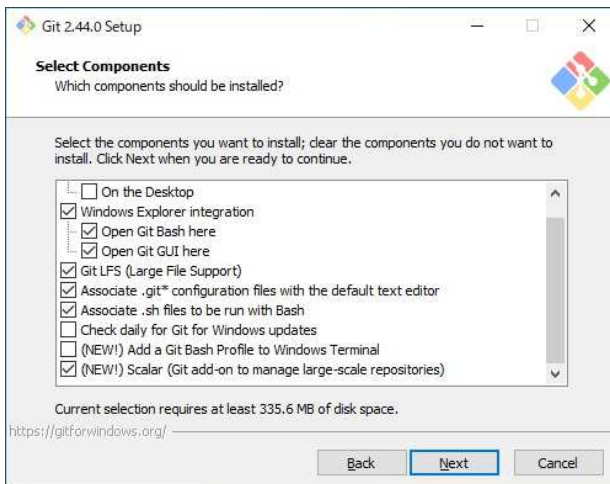
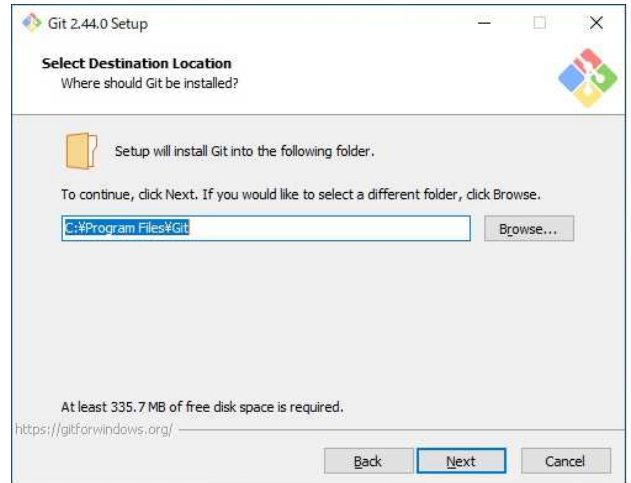


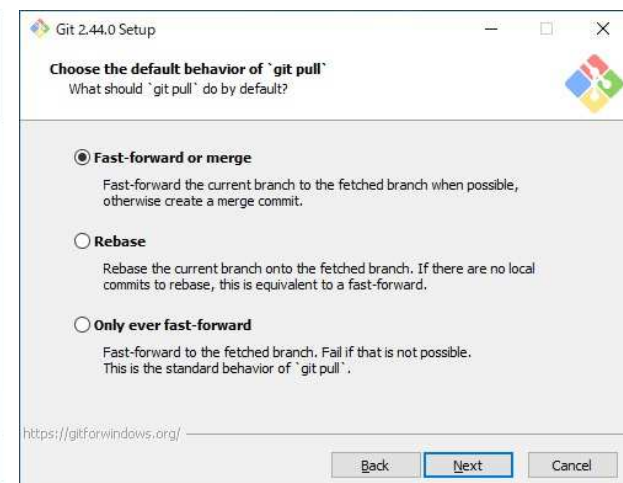
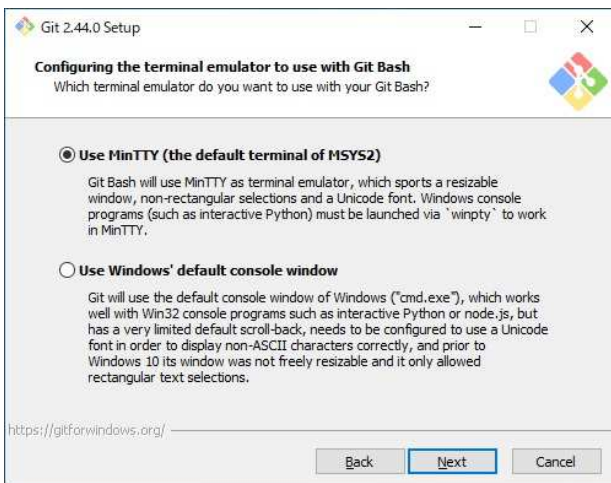
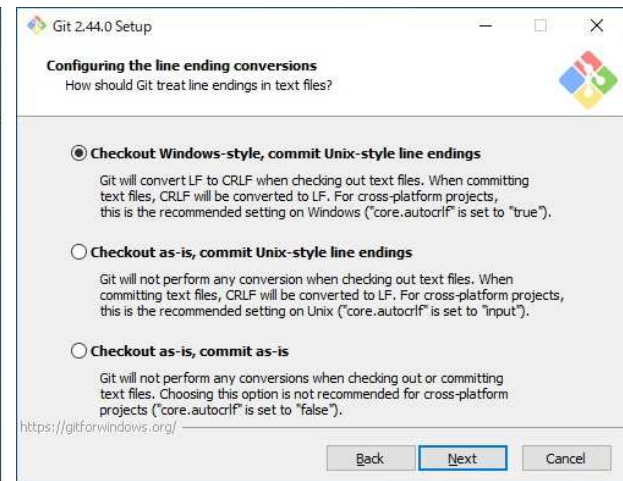
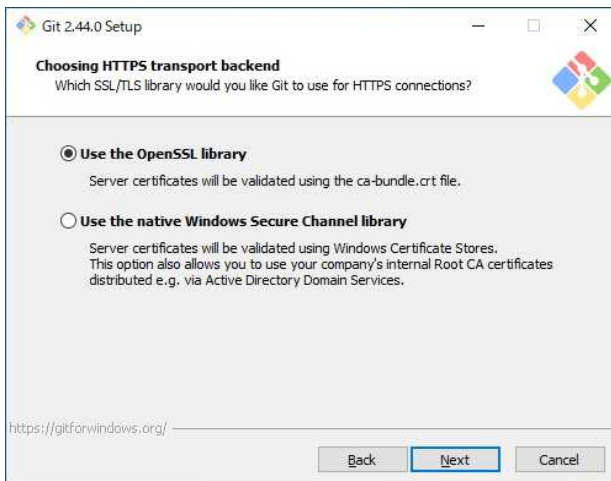
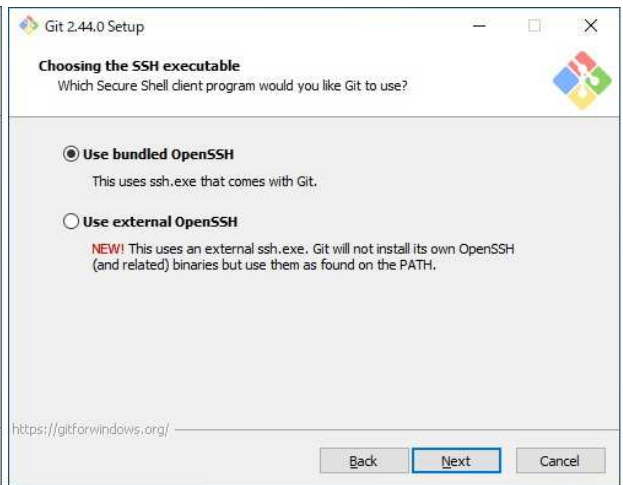
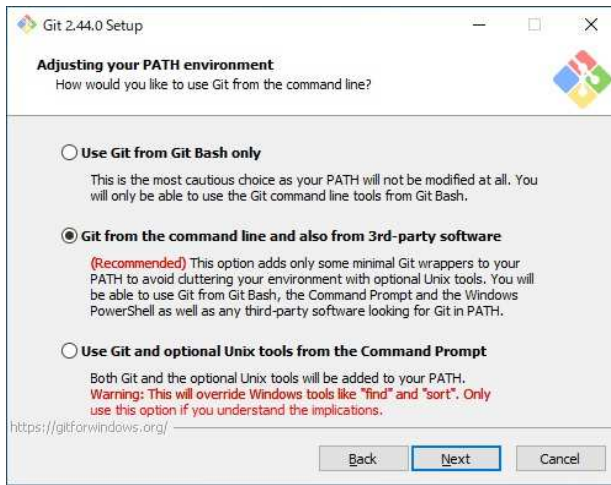
「Click here to download」を選択し、クリックする

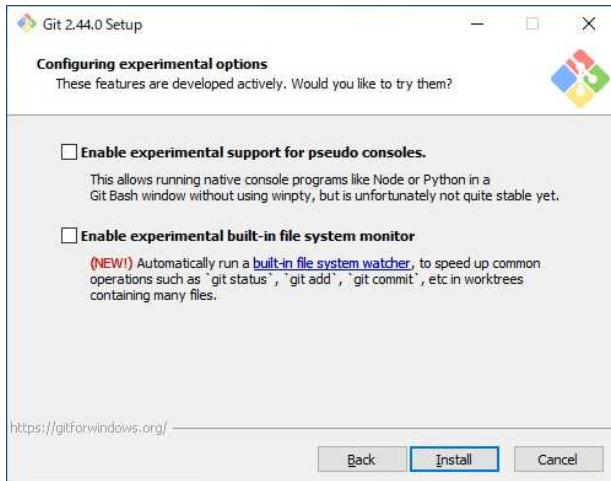
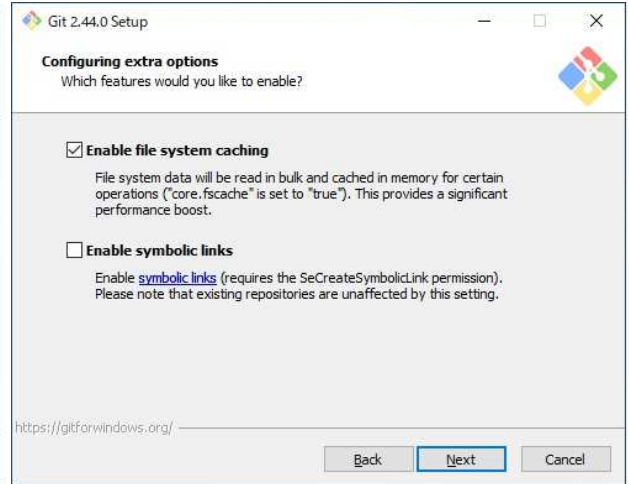
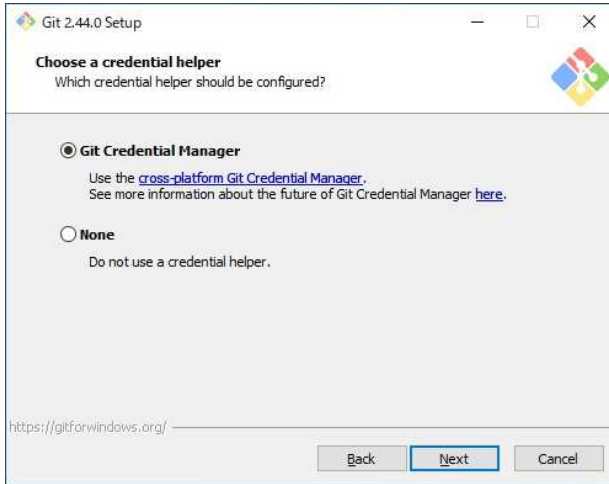


Git-2.44.0-64-bit をダウンロードする (バージョンは定期的に更新されている)

Git をインストール (Git-2.44.0-64-bit.exe をダブルクリック)  
【インストール画面】







ここまでくれば、git はインストール完了

## 6. YOLOv7 インストール

アプリから anaconda prompt を起動



コマンドを実行 (YOLOv7 の動作環境へ移行)

```
conda activate yolov7
```

```
Anaconda Prompt
(base) C:\Users\kimura>conda activate yolov7
(yolov7) C:\Users\kimura>_
```

YOLOv7 インストール

```
git clone https://github.com/WongKinYiu/yolov7.git
```

```
Anaconda Prompt
(base) C:\Users\kimura>conda activate yolov7
(yolov7) C:\Users\kimura>git clone https://github.com/WongKinYiu/yolov7.git_
```

```
Anaconda Prompt
(base) C:\Users\kimura>conda activate yolov7
(yolov7) C:\Users\kimura>git clone https://github.com/WongKinYiu/yolov7.git
Cloning into 'yolov7'...
remote: Enumerating objects: 1197, done.
remote: Total 1197 (delta 0), reused 0 (delta 0), pack-reused 1197
Receiving objects: 100% (1197/1197), 74.23 MiB | 20.93 MiB/s, done.
Resolving deltas: 100% (520/520), done.
Updating files: 100% (108/108), done.
(yolov7) C:\Users\kimura>_
```

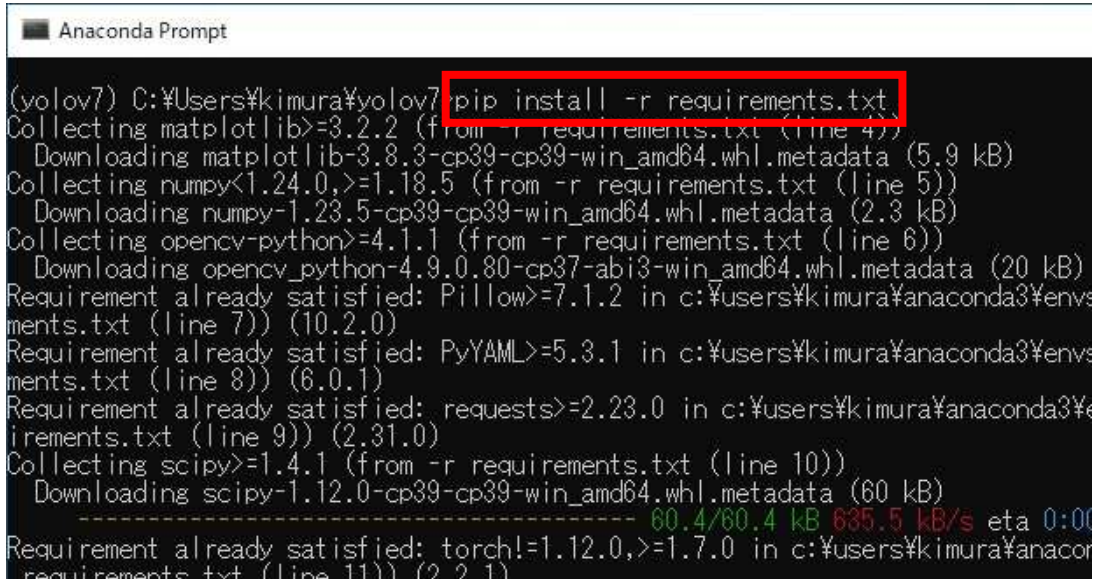
YOLOv7 実行フォルダの移動

```
cd yolov7
```

```
(yolov7) C:\Users\kimura>git clone https://github.com/WongKinYiu/yolov7.git
Cloning into 'yolov7'...
remote: Enumerating objects: 1197, done.
remote: Total 1197 (delta 0), reused 0 (delta 0), pack-reused 1197
Receiving objects: 100% (1197/1197), 74.23 MiB | 20.93 MiB/s, done.
Resolving deltas: 100% (520/520), done.
Updating files: 100% (108/108), done.
(yolov7) C:\Users\kimura>cd yolov7
(yolov7) C:\Users\kimura\yolov7>_
```

## YOLOv7 動作環境の復元

```
pip install -r requirements.txt
```

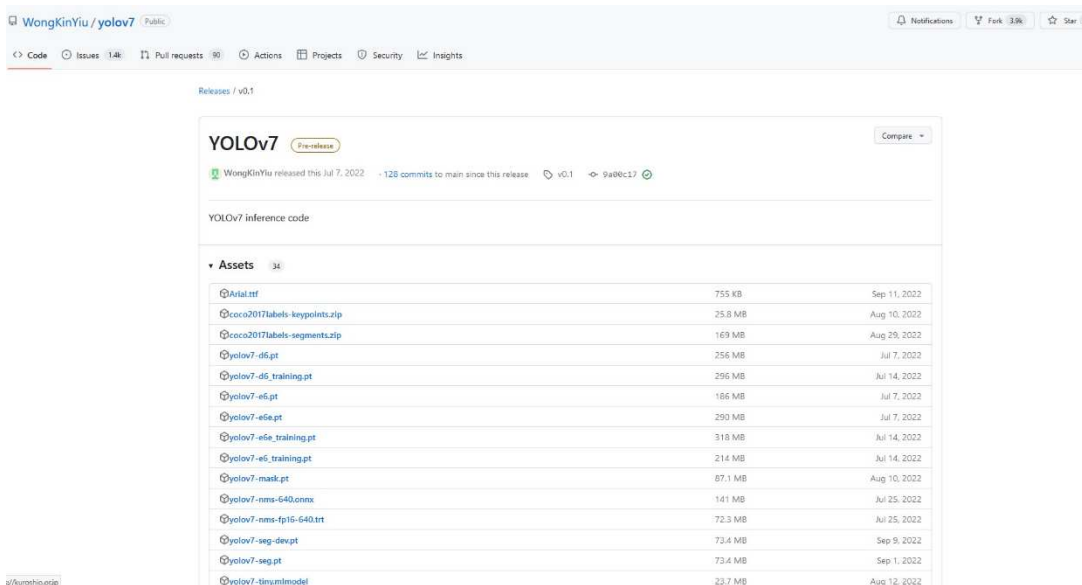


```
Anaconda Prompt
(yolov7) C:\Users\kimura\yolov7 pip install -r requirements.txt
Collecting matplotlib>=3.2.2 (from -r requirements.txt (line 4))
  Downloading matplotlib-3.8.3-cp39-cp39-win_amd64.whl.metadata (5.9 kB)
Collecting numpy<1.24.0,>=1.18.5 (from -r requirements.txt (line 5))
  Downloading numpy-1.23.5-cp39-cp39-win_amd64.whl.metadata (2.3 kB)
Collecting opencv-python>=4.1.1 (from -r requirements.txt (line 6))
  Downloading opencv_python-4.9.0.80-cp37-abi3-win_amd64.whl.metadata (20 kB)
Requirement already satisfied: Pillow>=7.1.2 in c:\users\kimura\anaconda3\envs\yolov7\lib\site-packages (from -r requirements.txt (line 7)) (10.2.0)
Requirement already satisfied: PyYAML>=5.3.1 in c:\users\kimura\anaconda3\envs\yolov7\lib\site-packages (from -r requirements.txt (line 8)) (6.0.1)
Requirement already satisfied: requests>=2.23.0 in c:\users\kimura\anaconda3\envs\yolov7\lib\site-packages (from -r requirements.txt (line 9)) (2.31.0)
Collecting scipy>=1.4.1 (from -r requirements.txt (line 10))
  Downloading scipy-1.12.0-cp39-cp39-win_amd64.whl.metadata (60 kB)
----- 60.4/60.4 kB 635.5 kB/s eta 0:00
Requirement already satisfied: torch!=1.12.0,>=1.7.0 in c:\users\kimura\anaconda3\envs\yolov7\lib\site-packages (from -r requirements.txt (line 11)) (2.2.1)
```

## YOLOv7 学習モデルのダウンロード

<https://github.com/WongKinYiu/yolov7/releases/tag/v0.1>

### 【ダウンロード画面】



WongKinYiu / yolov7 Public

Code Issues 14k Pull requests 90 Actions Projects Security Insights

Releases / v0.1

### YOLOv7 Pre-release

WongKinYiu released this Jul 7, 2022 · 128 commits to main since this release · v0.1 · 9a86c17

YOLOv7 inference code

#### Assets

Asset	Size	Created
Arial.ttf	755 KB	Sep 11, 2022
coco2017labels-keypoints.zip	25.8 MB	Aug 10, 2022
coco2017labels-segments.zip	169 MB	Aug 29, 2022
yolov7-d6.pt	256 MB	Jul 7, 2022
yolov7-d6_training.pt	296 MB	Jul 14, 2022
yolov7-e6.pt	186 MB	Jul 7, 2022
yolov7-e6e.pt	290 MB	Jul 7, 2022
yolov7-e6e_training.pt	318 MB	Jul 14, 2022
yolov7-e6_training.pt	214 MB	Jul 14, 2022
yolov7-mask.pt	87.1 MB	Aug 10, 2022
yolov7-nms-640.onnx	141 MB	Jul 25, 2022
yolov7-nms-fp16-640.tnt	72.3 MB	Jul 25, 2022
yolov7-seg-dev.pt	73.4 MB	Sep 9, 2022
yolov7-seg.pt	73.4 MB	Sep 1, 2022
yolov7-tinytmodel	23.7 MB	Aug 12, 2022



yolov7.pt をダウンロード

yolov7-w6-person.pt をダウンロード

 yolov7-w6-person.pt	134 MB
 yolov7-w6-pose.pt	154 MB
 yolov7-w6.pt	135 MB
 yolov7-w6_training.pt	158 MB
 yolov7.pt	72.1 MB
 yolov7.weights	141 MB
 yolov7x-seg.pt	140 MB
 yolov7x.pt	136 MB

yolov7.pt をダウンロード



yolov7-w6-person.pt をダウンロード



ダウンロードした2つのファイルを yolov7 フォルダに移動

学習モデルファイルの移動が完了すれば、YOLOv7 の動作環境は完成  
次は YOLOv7 の動作確認を行きましょう。

## 7. YOLOv7 検出テスト

YOLOv7 の検出テストを行う（コマンドを実行/1行で入力）

```
python detect.py --weights yolov7.pt  
--source inference/images/horses.jpg
```

```
(yolov7) C:\Users\kimura\yolov7>python detect.py --weights yolov7.pt --source inference/images/horses.jpg
```

（runs/detect フォルダの exp フォルダに結果がある）

（検出するたびに exp のあとに連番で数字が付加される）

（馬が検出できていれば、インストール完了）



※行動解析支援システムの利用環境が整いました※

