

公益財団法人
九州先端科学技術研究所

Institute of Systems,
Information Technologies
and Nanotechnologies

BODIK ODCSを活用した リアルタイム系オープンデータの可視化

2025年11月22日

ISIT 高野

- 福岡市の早良区役所と共同で取り組む西新駅駐輪場でのAI防犯カメラによる実証実験
- 利用者の入退場に基づくデータから、リアルタイムに駐輪場の混雑状況を可視化する仕組みを実装



西新駅中央駐輪場ご利用の皆さま

令和6年10月23日
早良区生活環境課
経済観光文化局企業連携課

mirai@共同事業『西新駅中央駐輪場の混雑状況をリアルタイムで判定』する実証実験を実施

mirai@city.fukuoka

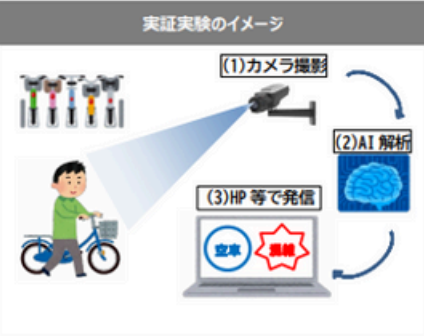
福岡市では、自転車の利用環境の向上などに取り組んでおりますが、この度、駐輪場利用者の利便性向上のため、画像認識 AI を活用し、駐輪場の混雑状況をリアルタイムで提供する実証実験を実施することとなりましたのでお知らせします。

1 実施期間
令和6年10月23日 ~ 令和6年11月30日

2 実施事業者
(公財)九州先端科学技術研究所

3 実証実験の概要等

実証実験のイメージ



概要

(1) 駐輪場の入退出者を、既設の防犯カメラで撮影

(2) 画像認識 AI にて撮影画像を解析・撮影した画像から自転車を検知し、AI による解析で駐輪場の混雑状況を判定
※検知後の画像データは削除

(3) 判定結果をホームページ等で発信
※ホームページ発信は11月上旬を予定

今知りたい「駐輪場」の満空情報を1タップで表示



西新駅中央駐輪場

2024/11/11 16時30分

混

混

1階

2階

満：満車に近い状況
混：入退場が多く混雑している状況
空：比較的空いている状況

昨日の混雑状況 (1階)

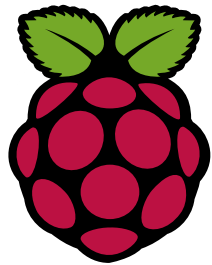
	昨日の時間帯	混雑状況
17.	午前10時00分	空
18.	午前10時15分	混
19.	午前10時30分	混
20.	午前10時45分	混
21.	午前11時00分	混
22.	午前11時15分	混
23.	午前11時30分	混
24.	午前11時45分	混
25.	午後12時00分	満
26.	午後12時15分	満

1 - 50 / 63

**課題：ある屋内空間の利用状況を分析・可視化して、
利用状況の把握・改善に役立てたい**



**PoCにお金をかけたらいけない。
なるべく安価に導入効果を計ってみよう。**



AIカメラはラズパイ5



検出データをBODIK ODCSにアップロード



GoogleのLooker Studioで可視化



AIカメラ一式（電源が必要）

AIカメラ（Raspberry Pi5+IMX500）



サイズ：約85（W） × 56（D） × 17（H） mm
重さ：100g以下
電源：ACアダプタ（5V5A）

電気代：月額100円強から260円程度

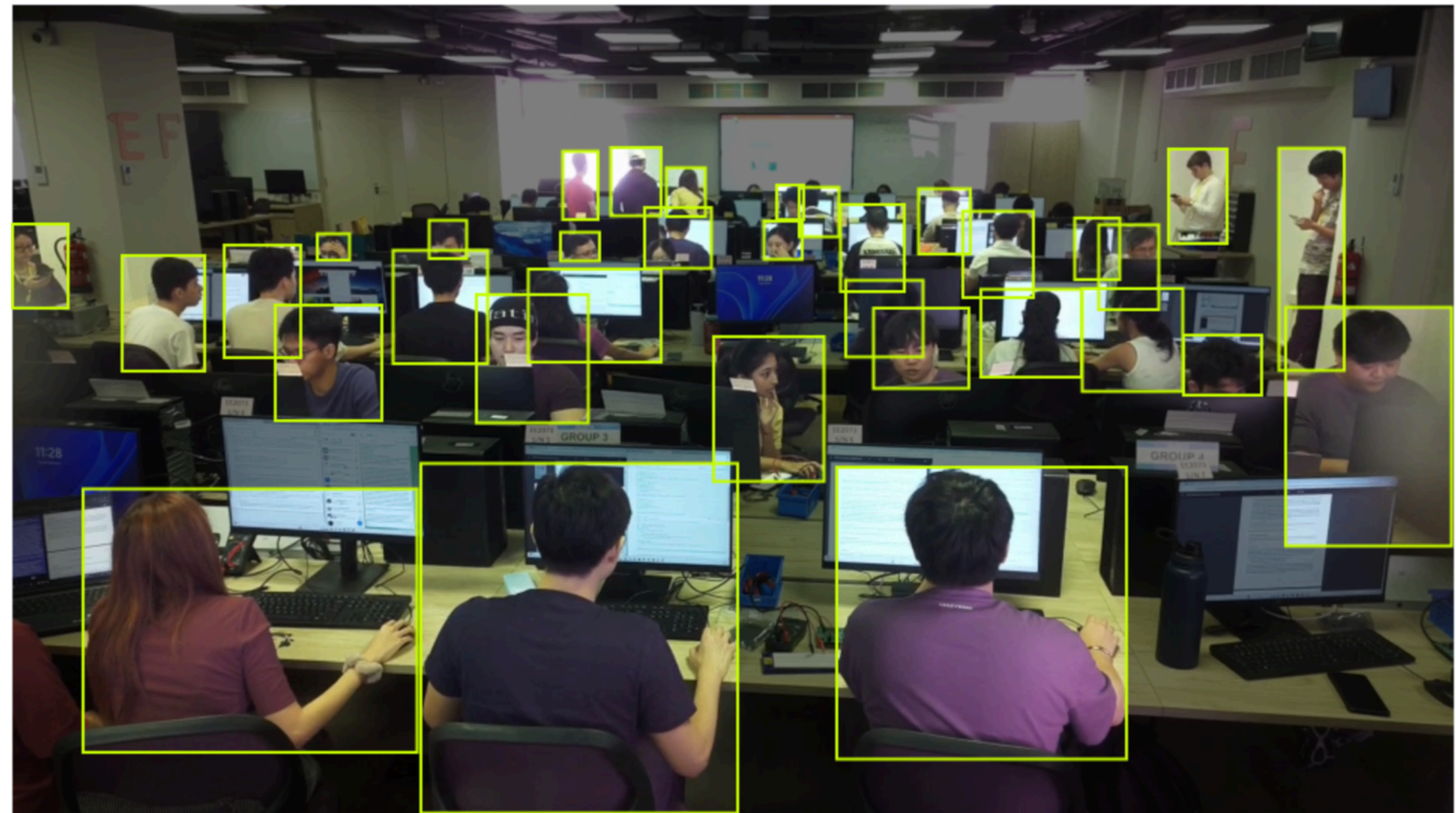
+

無線LANルータ（通信SIM）



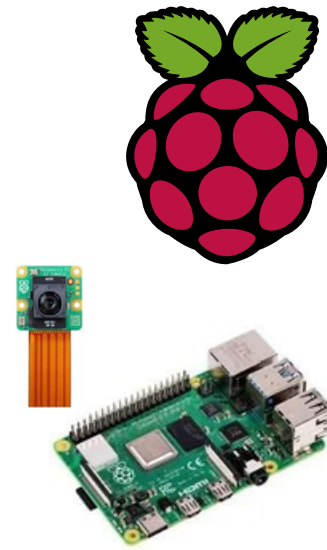
サイズ：約130(W) x 130(D) x 37(H) mm
重さ：約200g
電源：ACアダプタ

- AIカメラ一式を教室の後方（全体が見下ろせる画角）などに設置
- 運用時の映像は即削除（検知数のみISITサーバに送信）



検知イメージ

- AIカメラの機能（人の検知）をラズパイに実装



ラズパイをセットアップ
難易度：★



Yoloベースの
物体検出プログラム
&
データ送信プログラム
難易度：★★★★～

- AIカメラの機能（人の検知）をラズパイに実装

検知項目は、日時、ラベル、クラス名、確からしさ、矩形情報
filename,label,class_id,confidence,x_min,y_min,x_max,y_max,width,height
20251121_1300.png,person,0,0.8462375402450562,63,423,267,763,204,340
20251121_1300.png,person,0,0.8231611251831055,334,461,550,825,216,364
20251121_1300.png,person,0,0.7157149910926819,302,353,393,500,91,147
20251121_1300.png,person,0,0.6929333209991455,3,358,102,469,99,111
20251121_1300.png,person,0,0.6516324877738953,702,375,801,465,99,90
20251121_1300.png,person,0,0.614531934261322,568,392,748,573,180,181
20251121_1300.png,person,0,0.5631160736083984,401,392,531,523,130,131
20251121_1300.png,person,0,0.5253376364707947,669,471,928,729,259,258
20251121_1300.png,person,0,0.5128548741340637,766,355,842,449,76,94
20251121_1300.png,person,0,0.46445411443710327,658,365,715,440,57,75
20251121_1300.png,person,0,0.4512734115123749,813,389,952,529,139,140
20251121_1300.png,person,0,0.400265097618103,407,354,468,430,61,76
20251121_1300.png,person,0,0.3994739353656769,211,356,280,447,69,91



一定時間間隔で検知データを集約

データ項目は、トータルカウント数、人の検知数

datetime,total_count,person
2025-11-21 13:00:00,13,13
2025-11-21 13:10:00,13,13

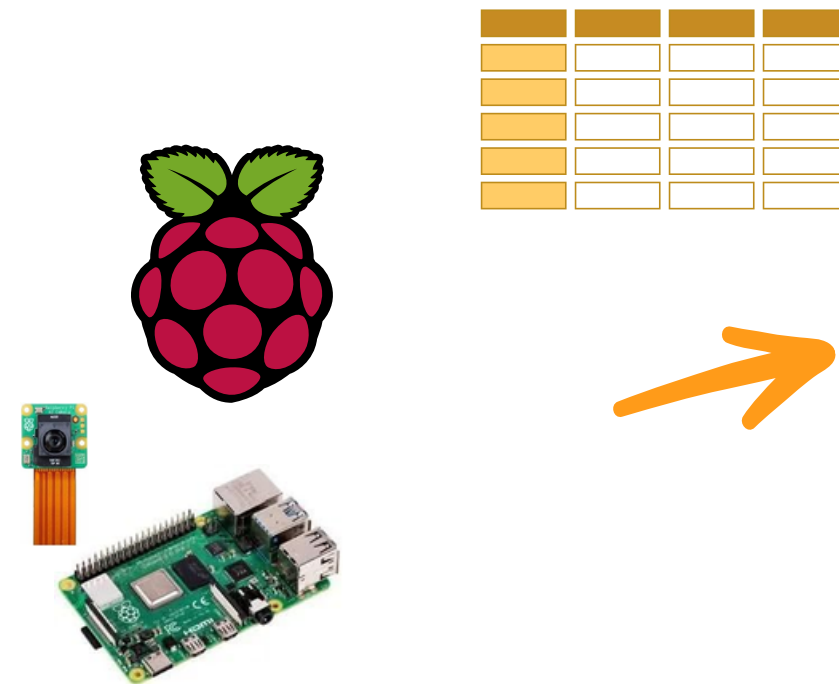
※ このとき、映像は削除します。



※ 元の画像はGeminiで生成しました

- CKANサーバにデータを自動でアップロードする機能を実装

集約された検知データ（csv）



**BODIK ODCS等
CKANサーバ**

難易度：★（自治体職員）

○分毎に自動送信
難易度：★★



**個人・民間の場合、中間サーバが必要
APIを叩いてサーバレスにする方法も**

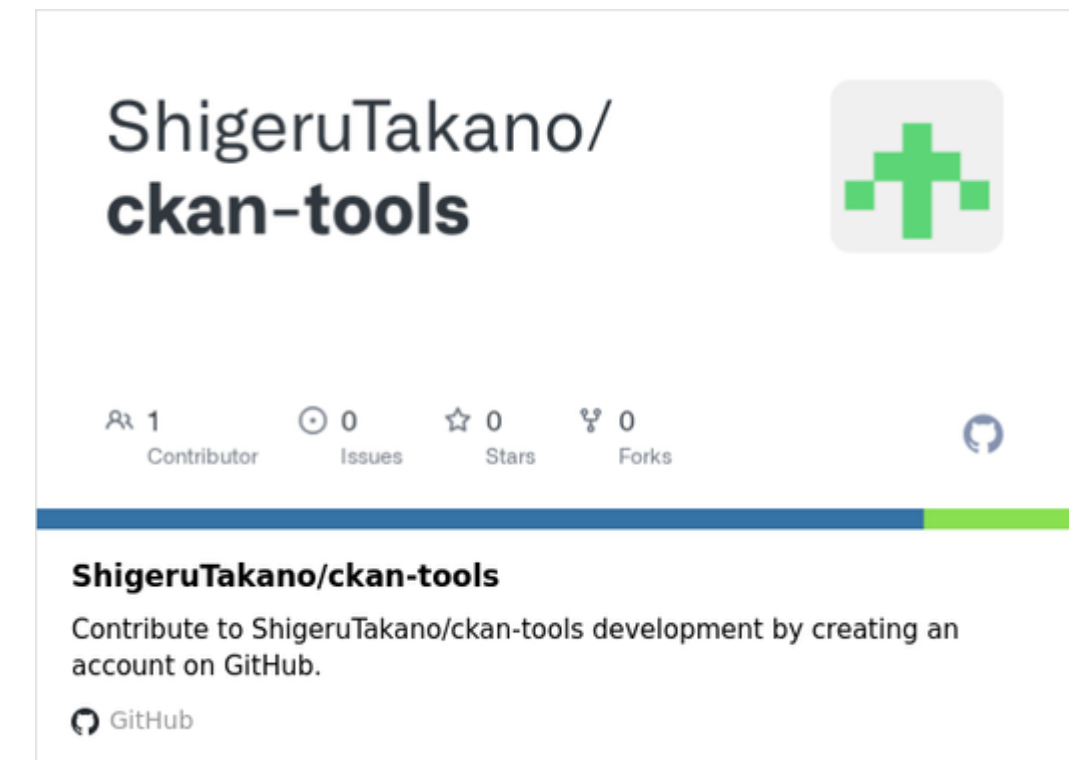
難易度：★★★★★

- CKANにデータを自動で送信

CKANにデータセットを作成し、初期リソース（csv）を登録



CKANの操作ツール



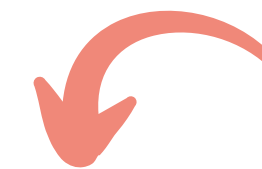
CKANサーバにデータをアップロードするツールで自動化

名前で作成・更新

```
python ckan_resource_cli.py --ckan-url <CKANのURL> upload <api_key> <package_id>
<file_path> --resource-name <リソース名> [--description <説明>]
```

IDで更新

```
python ckan_resource_cli.py --ckan-url <CKANのURL> upload <api_key> <package_id>
<file_path> --resource-id <リソースID> [--description <説明>]
```

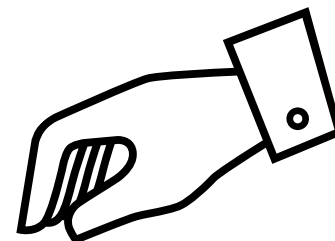
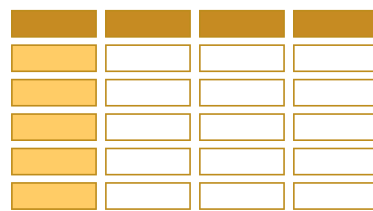


※ BODIK ODCSの場合、
API トークンを発行して利用しましょう

- CKANサーバにあるデータを自動でスプレッドシートに転記&可視化

GASでCSVデータを取得し、
スプレッドシートに展開

難易度：★★★★



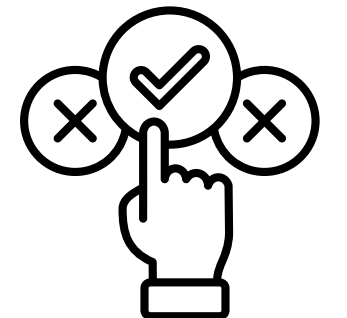
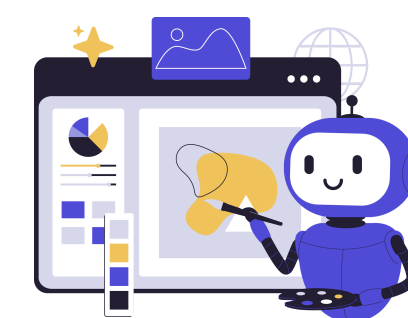
BODIK ODCS等
CKANサーバ

Googleスプレッドシート



Looker Studioで可視化

難易度：★~★★★★★★



AIが作ったものを
人間が選択する時代？ 9

- CKANサーバにあるデータを自動でスプレッドシートに転記

Apps Scriptの例（CKANのCSVデータを読み込んでスプレットシートに転記する）

```
function myFunction() {
  // CKANのリソース情報
  var resource = 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX';
  var limit = 3600;
  var requestUrl = `https://data.bodik.jp/api/3/action/datastore_search?resource_id=${resource}&limit=${limit}`;
  // APIキーをリクエストヘッダーに追加
  var options = {
    "headers": {
      "Authorization": "API TOKEN" // CKANのAPIキーをここに入力
    }
  };
  // CKANからデータを取得
  var response = UrlFetchApp.fetch(requestUrl,options);
  var res = JSON.parse(response.getContentText()).result.records;
  // データ項目を取得
  var values = res.map((content) => {
    return [content['datetime'], content['total_count'], content['person']]
  });
  // データの項目名をセット
  var title = [['datetime', 'total_count', 'person']];
  // 項目名とデータを結合する
  values = [...title, ...values];
  // 結合したデータをスプレッドシートに出力する
  var sheet = getOrCreateSheet("シート1");
  // シートの内容をすべてクリアして書き込み
  sheet.clear();
  sheet.getRange(`A1:C${values.length}`).setValues(values);
}
```

AIカメラのデモ ☆ 📁 ☁

ファイル 編集 表示 挿入 表示形式 データ ツール 拡張機能 ヘルプ

🔍 ⏪ ⏩ 🖨 📄 100% ▼ | ¥ % .0 .00 123 | デフォ... ▼ | - 10

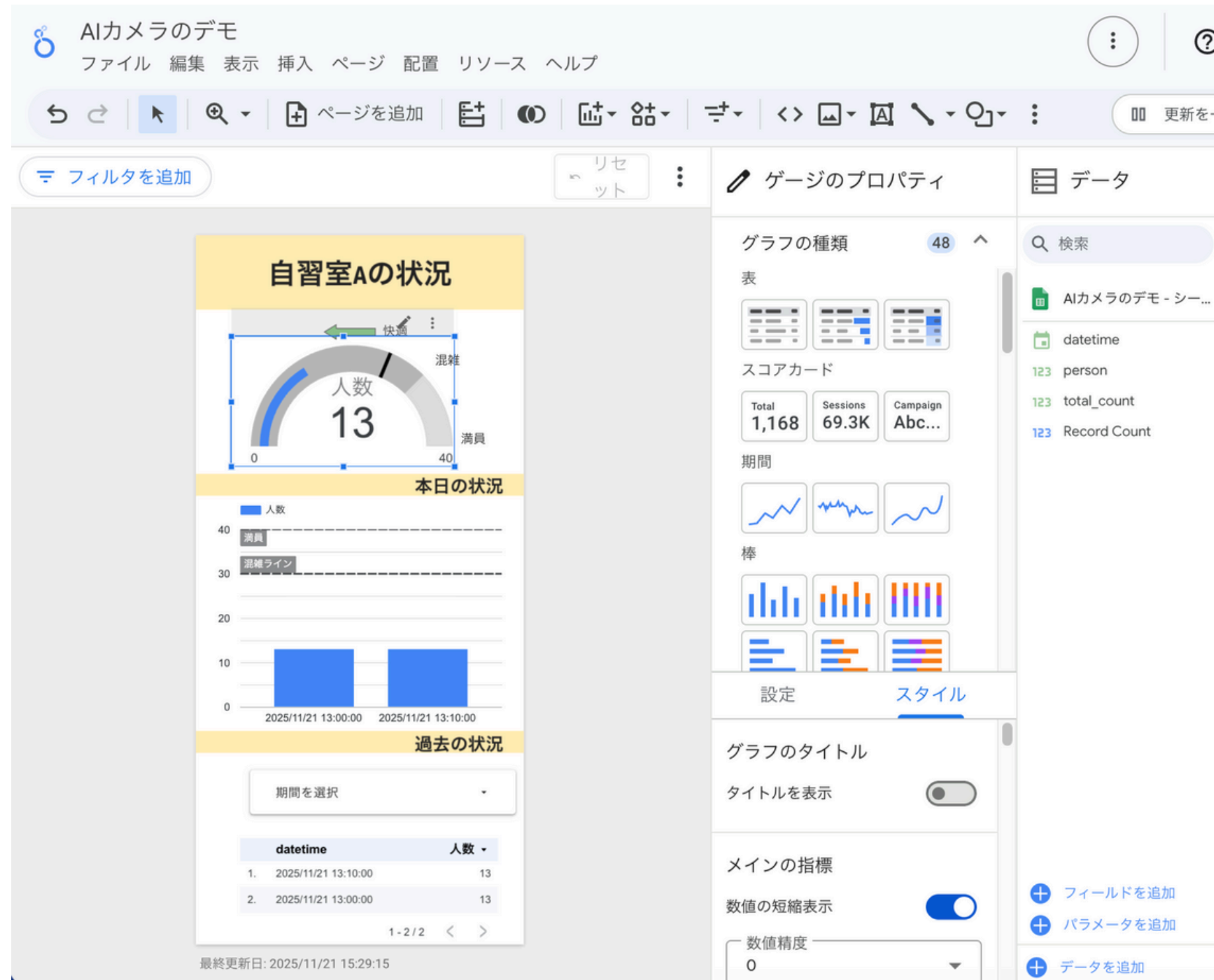
13 ▼ | fx

	A	B	C	D	E
1	datetime	total_count	person		
2	2025-11-21 13:00:00	13	13		
3					
4					
5					
6					
7					
8					

時間や更新時に自動的に更新するようにトリガーを設定する

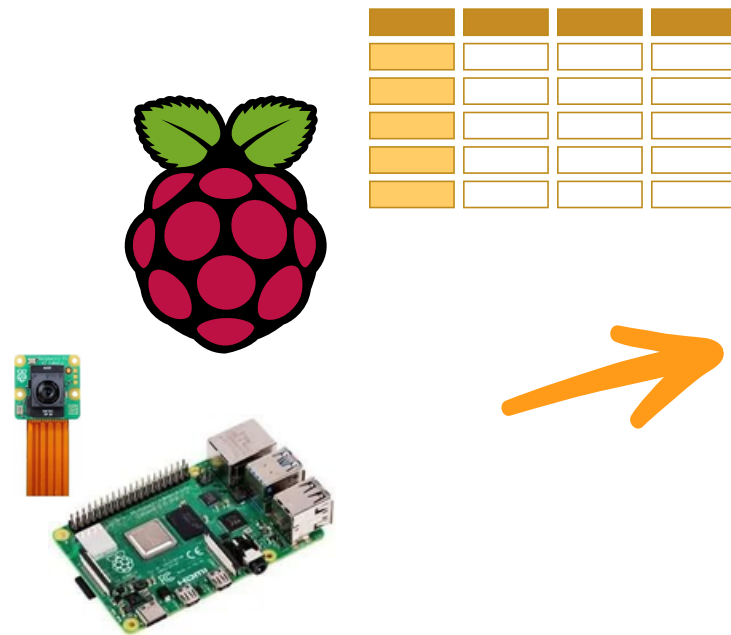
- スプレッドシートのデータを元にLooker Studioのアプリを実装

ここまでくれば人間でも
10分くらいでアプリを作成・公開できます！



データの流れ

集約された検知データ (csv)



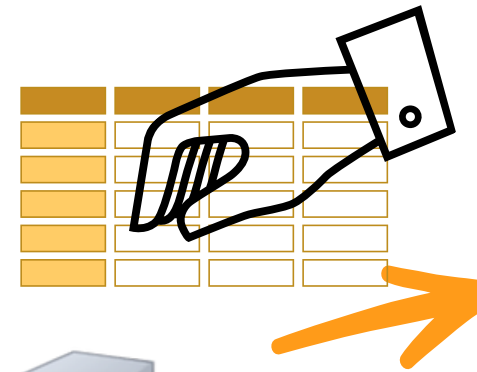
ラズパイをセットアップ



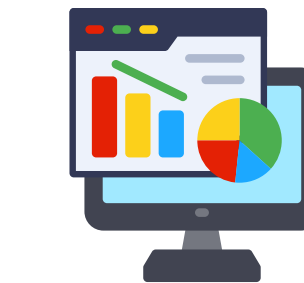
Yoloベースの
物体検出プログラム
&
データ送信プログラム



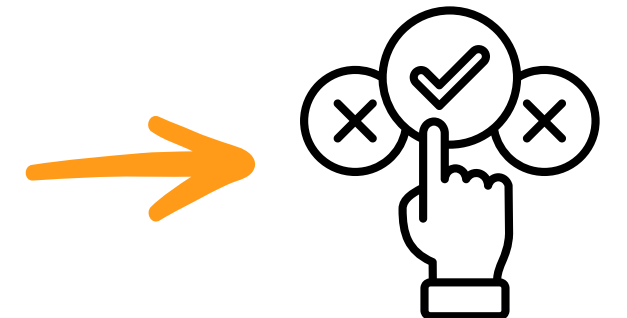
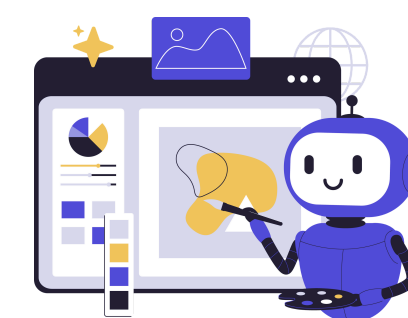
BODIK ODCS等
CKANサーバ



Google
スプレッドシート



Looker Studioで可視化



AIが作ったものを
人間が選択する時代？



- BODIK ODCSを活用して、ラズパイとカメラを置くだけで、混雑状況を可視化するところまで自動化する方法を公開
- 駐車場や道路カメラでも応用可能
- 今後、BODIK City OS（都市OS）の取り組みを加速



今知りたい「施設」の混雑情報を1タップで表示

