選別ラインのゴミを
画像で見分ける技術検討

公開データ（WaRP）と MATLAB での
分類・検出・輪郭の試作

2026年11月　5日

株式会社システムラボラトリ

本資料の概要

- 目的: 資源ごみの選別工場のコンベア画像で、5 種類（PETボトル・プラ容器・びん・缶・紙パック）を見分けられるかを MATLAB で確かめる。
- 1 個ずつ切り出した画像の分類は、正解率 88.1% である。
- 正解の量・撮影の日・画質を変えると、条件によっては 76〜79% まで落ちる。強くぼかすと 64.7% である。
  - 落ち方は種類で違う。数の少ないびん・缶、暗さに弱い缶・びん・紙パックから崩れる。
- コンベアの画像から場所と種類を当てる検出は、AP50 の平均 0.43 である。80 エポックまで学習を延ばしても、これを超えなかった。
  - 検出の枠を SAM に渡すと、物ごとの輪郭まで切り出せる。正解の輪郭との重なりは 0.59→0.69 に上がる。
- 普通のカラー画像では、樹脂の種類や電池の混入は見分けられない。何を見分けたいかで、センサーを先に選ぶ必要がある。

本検討で作ったもの

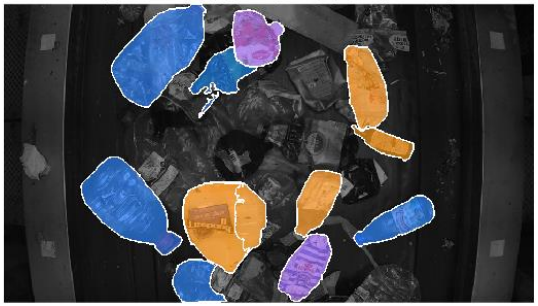
1. コンベアの写真



2. AI が見つけて種類を当てる（検出）



3. 物ごとの輪郭を切り出す（SAM）



コンベアの写真から、AI が資源ごみを見つけて種類（PETボトル・プラ容器・びん・缶・紙パック）を当て、SAM で物ごとの輪郭まで切り出す。

使ったのは、公開されている選別工場の画像と MATLAB、ノート PC だけである。

問題設定: 何を解くのか

選別の向き	何をするか	例	今回
拾い出し	混ざって流れてくるごみの中から、資源になる物を見つけて、種類ごとの箱に移す。	缶・びん・ペットボトルが混ざった中から、ペットボトルを拾う。	対象（その手前）
取り除き	ほぼ 1 種類だけが流れる線から、混ざった異物を抜いて純度を上げる。	紙だけを残したい線から、プラスチックや金属を抜く。	対象外

今回解くのは、混ざって流れてくる中から、資源になる物が「どこにあって」「何なのか」を画像で当てることである。人やロボットが拾って振り分ける前段に当たる。

5 種類は、振り分け先（日本の家庭ごみの分別の箱）に合わせてまとめた。

缶は磁石（鉄）や渦電流（アルミ）でも分けられる。画像で見分ける価値が大きいのは、ペットボトルとそれ以外のプラスチック容器の見分けや、異物の取り除きの方と考える。

使ったデータ（WaRP）

5 種類	元の 28 種類（WaRP の分類名）	学習	試験
PETボトル	透明・青・緑・茶・多色・油のボトルなど（潰れた物・空気入りを含む）14 種類。	4,759	911
プラ容器	乳製品の白いボトル・洗剤のボトル・樹脂の箱・ポリタンク 8 種類。	1,621	294
びん	透明・緑・茶のガラスびん 3 種類。	399	86
缶	缶 1 種類。	508	98
紙パック	飲料・牛乳の紙パック 2 種類。	584	162

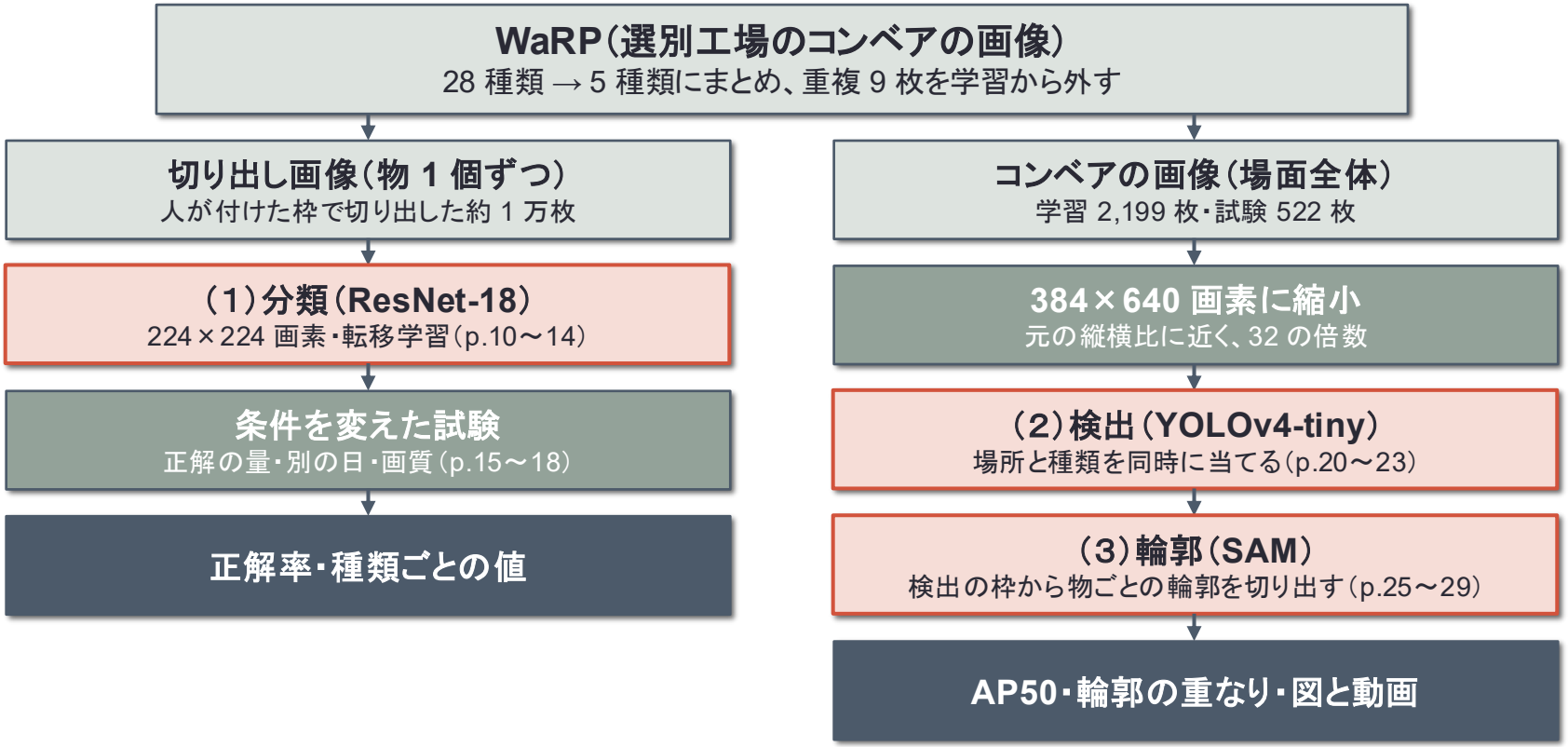
WaRP（Waste Recycling Plant Dataset）は、資源ごみの選別工場のコンベアを真上から撮った公開データ（Kaggle で CC0）。画像は 1920 × 1080 と 960 × 540。

数は物の数（人が付けた枠の数）。日本の家庭ごみの分別に合わせて 28 種類を 5 種類にまとめた。PETボトルが 6 割、びんは 5% と偏っている。

データを調べて分かった注意点

- 学習用と試験用に、まったく同じ画像が 9 枚入っていた。
  - ファイル名もバイト列も同一。学習用から外した。
- 試験用の画像 522 枚のうち 280 枚（54%）が、学習用のどれかと 30 秒以内に撮られていた。
  - ファイル名の撮影時刻で比べた。2 分以内では 467 枚、中央値は 27 秒。
  - コンベアは動いているので同じ物とは限らないが、日・照明・カメラは同じである。
- このため、公式の分け方で測った精度は甘めに出ると考える。
  - 確かめるために、試験の 57% を占める 3 月 25 日の画像を学習から外して試した（p.15〜16）。
- README と実物の違いもあった。分類の試験用は 1,583 枚ではなく 1,551 枚、画像は 1920×1080 だけでなく 960×540 も混ざる。

検討の全体の流れ



左が（1）分類（1 個ずつ切り出した画像に種類を付ける）。右が（2）検出（場面全体から物の場所と種類を当てる）と、その枠を使う（3）輪郭。

動作環境

項目	内容
ソフトウェア	MATLAB R2025b（Deep Learning Toolbox、Computer Vision Toolbox、Image Processing Toolbox）。
学習済みの AI	ResNet-18（分類）、YOLOv4-tiny（検出・COCO で学習済み）。どちらも MathWorks 提供の物。
PC	ノート PC 2 台。GPU は NVIDIA GeForce RTX 3050 Laptop（メモリ 4GB）と RTX 4050 Laptop（6GB）。
学習時間（分類）	333 秒（8 エポック・1 エポック 42 秒・RTX 3050）。
学習時間（検出）	1 回目 1,085 秒（RTX 3050）＋続き 1,083 秒（RTX 4050）。どちらも 18 分で打ち切り。比べた 80 エポックの学習は 8,788 秒（RTX 3050）。
データの準備	zip の展開 44～79 秒、検出用の縮小画像の作成 54～104 秒（PC により異なる）。

学習時間は 1 回の実測。検出は RTX 4050 でも 1 エポックが短くならなかった（121 秒→135 秒）ので、遅いのは GPU ではなく CPU での画像の読み込みと水増しと考える。

（1）分類

物を 1 個ずつ切り出した画像に、種類を付ける
AI: ResNet-18
正解の量・撮影の日・画質を変えた試験も含む

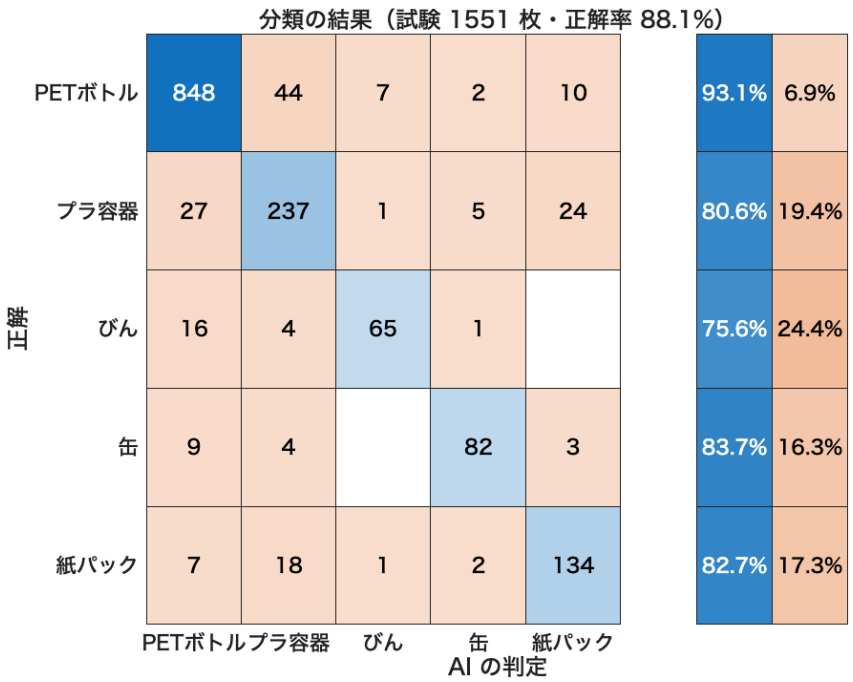
分類(1)方針

項目	内容
出発点	ResNet-18(画像の分類でよく使われる小さめの AI)。最後の層を 5 種類に付け替えて全体を学び直す(転移学習)。
入力	切り出し画像を 224×224 画素に揃える。
水増し	上下左右の反転・回転・拡大縮小(コンベアを真上から見るので、向きに意味は無い)。
学習条件	adam・学習率 1e-4・ミニバッチ 64・8 エポック。
枚数	学習 7,871 枚・検証 896 枚・試験 1,551 枚(試験は公式の試験用)。

コードの要点(MATLAB。学習の 3 行)。

```
net = imagePretrainedNetwork("resnet18", NumClasses=5);
options = trainingOptions("adam", InitialLearnRate=1e-4, MaxEpochs=8,
MiniBatchSize=64);
net = trainnet(dsTrain, net, "crossentropy", options);
```

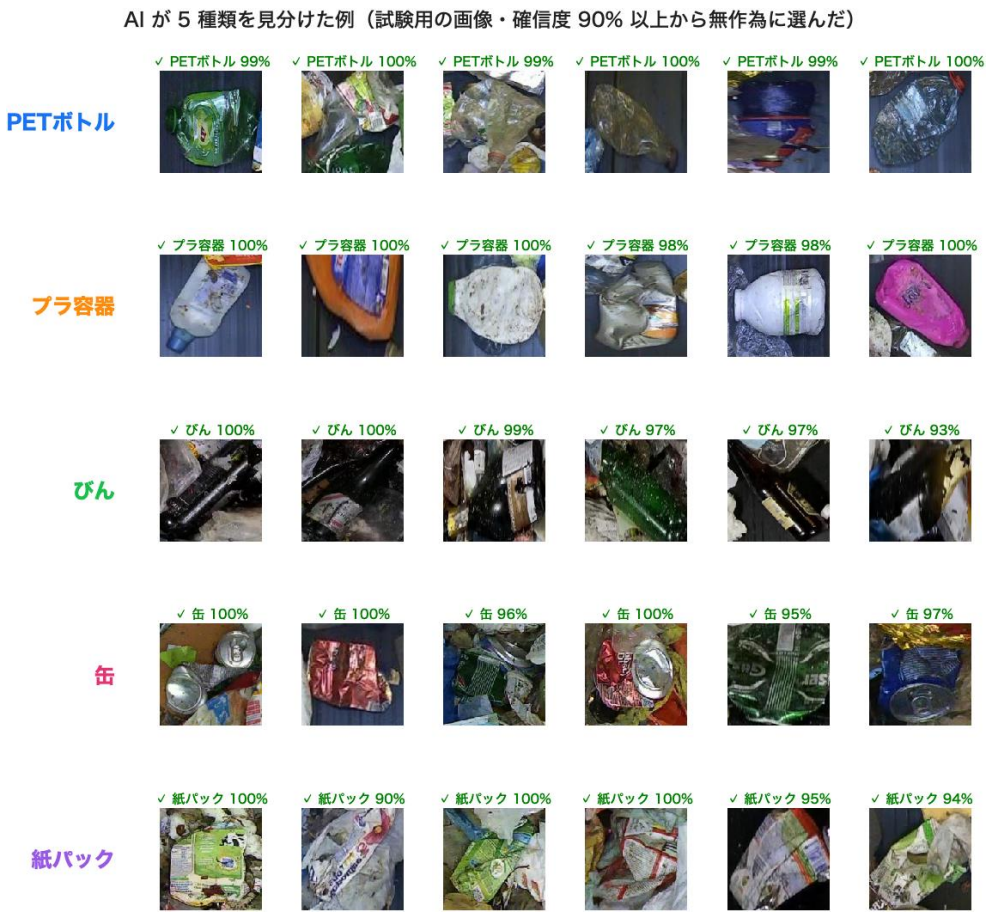
分類(2)結果



試験 1,551 枚で正解率 88.1%、5 種類の平均 F1 は 0.840。行が正解、列が AI の判定、右の列は種類ごとの当たり・外れの割合。

上図より、取り違えは物の性質どおりである。潰れた PETボトル→プラ容器（44 件）、緑・茶の PETボトル↔びん（びんの 16 件が PETボトルに）、乳製品の白いボトル↔牛乳の紙パック（24 件と 18 件）。

分類(3)AI が見分けた例



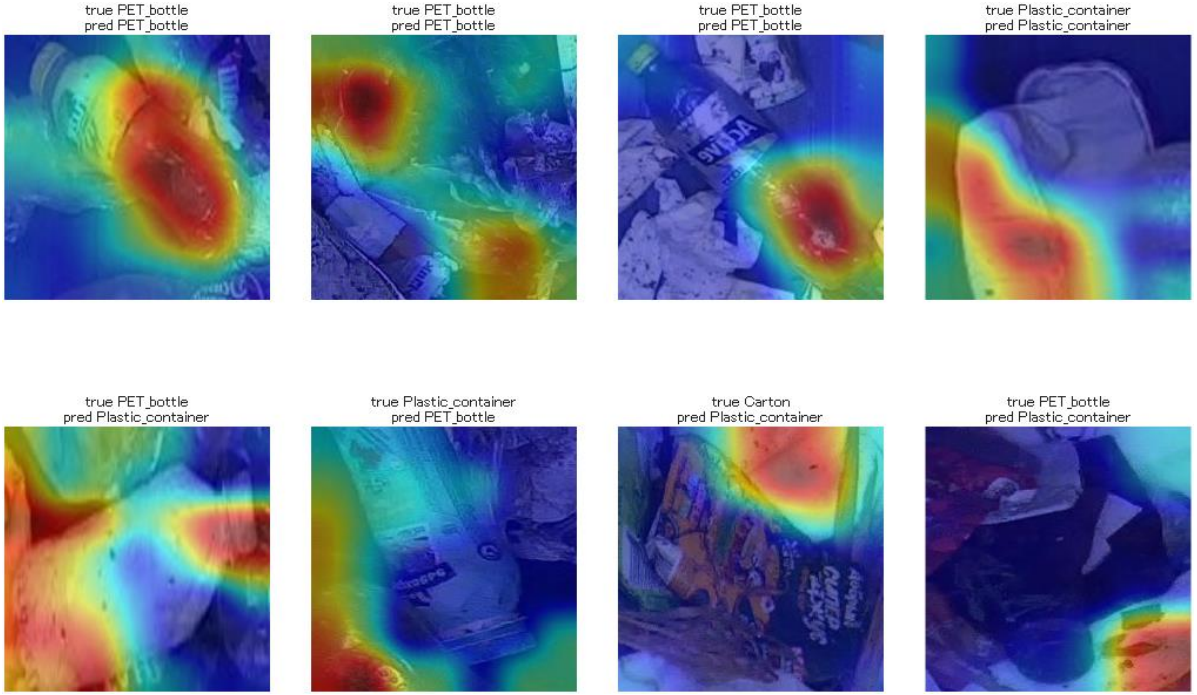
試験用の画像で、AI の確信度が 90% 以上だったものから種類ごとに無作為に 6 枚選んだ。潰れたボトル・汚れた紙パックも見分けている。

分類(4)AI が間違えた例



試験用の画像で AI が間違えたものから無作為に 12 枚選んだ。
白い乳製品ボトルと紙パック、潰れた PETボトルとプラ容器のように、人が見ても迷う組み合わせが多い。

分類(5)AI が見ていた場所



Grad-CAM(AI が判断に使った場所を色で重ねる方法)。赤いほど判断に効いた場所。上段は当たった例、下段は外れた例。

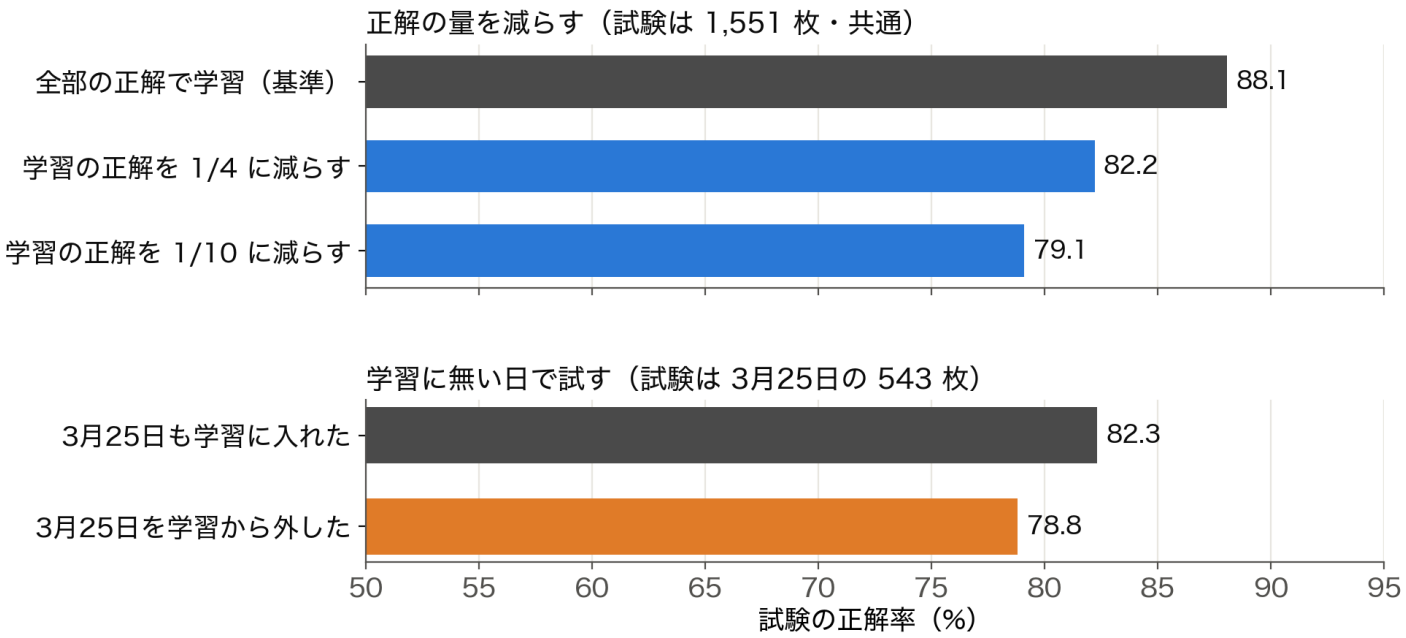
上図より、外れた例では、切り出し画像に写り込んだ隣の物を見ていることがある。1 個だけ置いた写真と、物が重なるコンベアの違いがここに出る。

条件を変えた試験（1）方針

条件	やり方	確かめたいこと
正解の量	学習の切り出し画像を、種類ごとに同じ割合で 1/4・1/10 に減らして学び直す（8 エポック）。	正解を付ける手間を減らしても使えるか。
別の日	試験の 57% を占める 3 月 25 日の画像を学習から外して学び直し、その日の試験画像（543 個）だけで試す。	学習に無い日の画像でどれだけ落ちるか。
画質	学習はそのまま。試験画像だけを暗く（明るさ 1/2・1/4）、ぼかす（弱・強）、解像度を下げる（1/2・1/4）。	照明・ぶれ・画素数のどれが効くか。

各条件の学習は 1 回ずつ（乱数を変えた繰り返しはしていない）。
正解の量を減らした条件はエポック数を揃えたので、枚数が少ないほど学習の回数も少ない。

条件を変えた試験(2)学習の画像を変えた結果

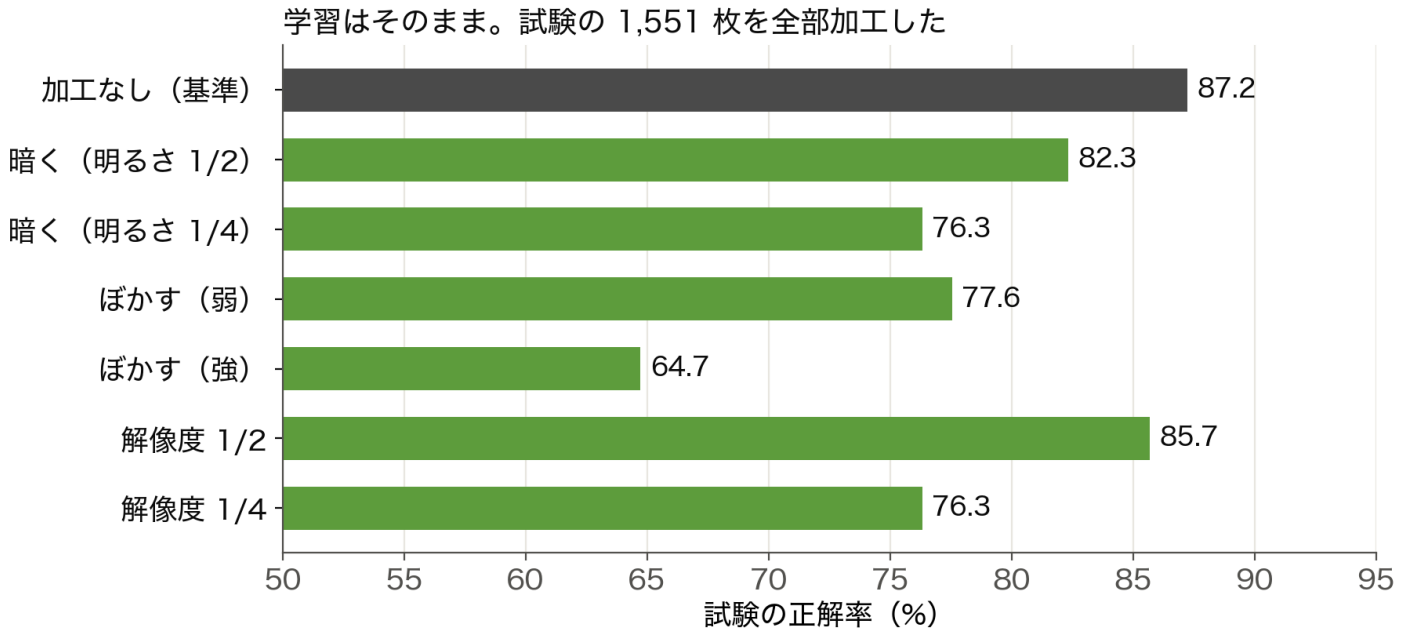


正解の量を減らした条件の試験は共通の 1,551 枚。別の日の条件の試験は 3 月 25 日の 543 個で、比べる基準はその日も学習に入れたときの 82.3%。

上図より、正解を 1/10 にしても 79.1%、学習に無い日の画像では 78.8% と、全体の正解率は緩やかに落ちる。

ただし正解を 1/10 にすると、びんの F1 は 0.81→0.51、缶は 0.86→0.63 と、数の少ない種類から崩れる（PETボトルは 0.93→0.88）。

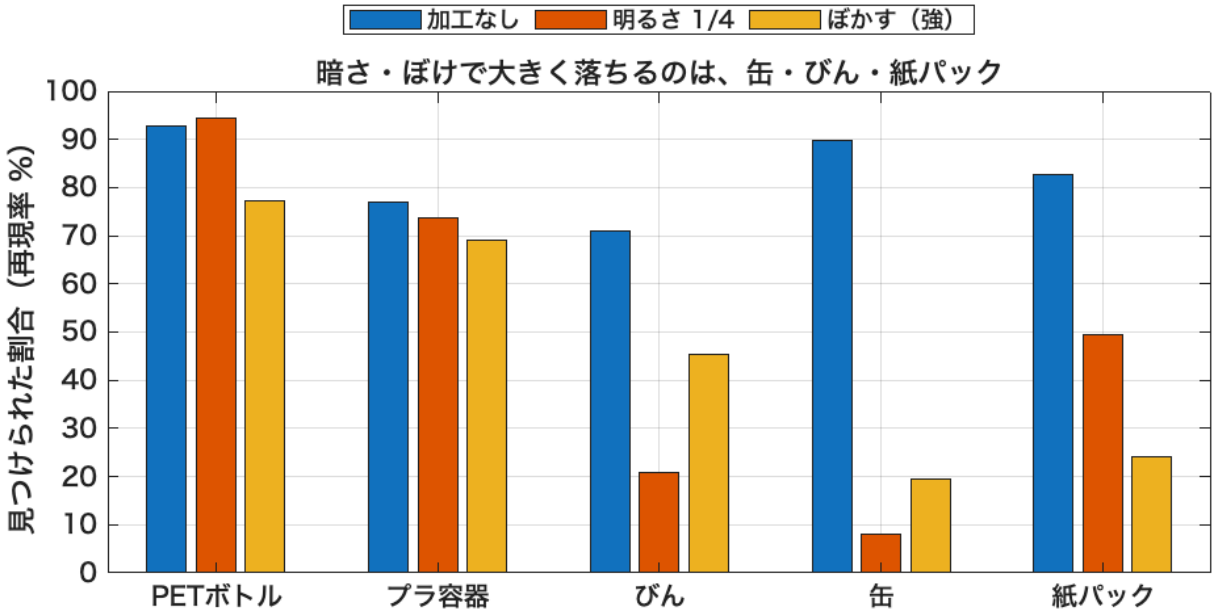
条件を変えた試験(3)試験の画像を悪くした結果



学習はそのままで、試験の 1,551 枚を全部加工した。基準の「加工なし」は 87.2%（全体の 88.1% と少し違うのは、評価の経路で画像を縮める補間の方法が違うため）。

上図より、強くぼかすと 64.7% まで落ちる。解像度は 1/2 ならほとんど落ちない（85.7%）。カメラの画素数より、照明とぶれを整える方が効くと考える。

条件を変えた試験(4)種類ごとの落ち方



試験の画像を暗く・ぼかしたときの、種類ごとの見つけられた割合(再現率)。

缶は暗くすると、見つけられた割合が 90% から 8% に落ちる。びんも 71% から 21% である。一方 PETボトルは落ちない(迷うと数の多い PETボトルと答える傾向があるとみる)。

上図より、全体の正解率だけでは崩れ方に気づけない。種類ごとの値で判断する必要がある。

（2）検出

コンベアの場面全体から、物の場所と種類を同時に当てる
AI: YOLOv4-tiny

検出(1)方針

項目	内容
出発点	YOLOv4-tiny(軽い検出器。COCO の 80 種類で学習済み)を 5 種類に学び直す。
入力	384×640 画素(元の 1920×1080 の縦横比に近く、32 の倍数)。
出力	物を囲む長方形の枠(外枠。バウンディングボックス)と、その種類・確信度。
枠の形の候補	学習データの枠から 6 個を推定(アンカーボックス。平均 IoU 0.76)。
水増し	上下左右の反転と、明るさ・コントラスト・彩度のゆらぎ。
学習条件	1 回目: adam・学習率 1e-3・ミニバッチ 8。続き: 学習率 1e-4。どちらも 18 分で打ち切り(約 8.5＋8 エポック)。80 エポックの学習とも比べた(p.22)。
枚数	学習 2,199 枚(物 7,871 個)・試験 522 枚(物 1,551 個)。

MATLAB R2025b の RTMDet(別の検出器)は、学習済みの 80 種類で使う専用で、5 種類への学び直しには使えなかった。そのため YOLOv4-tiny にした。

検出(2)結果

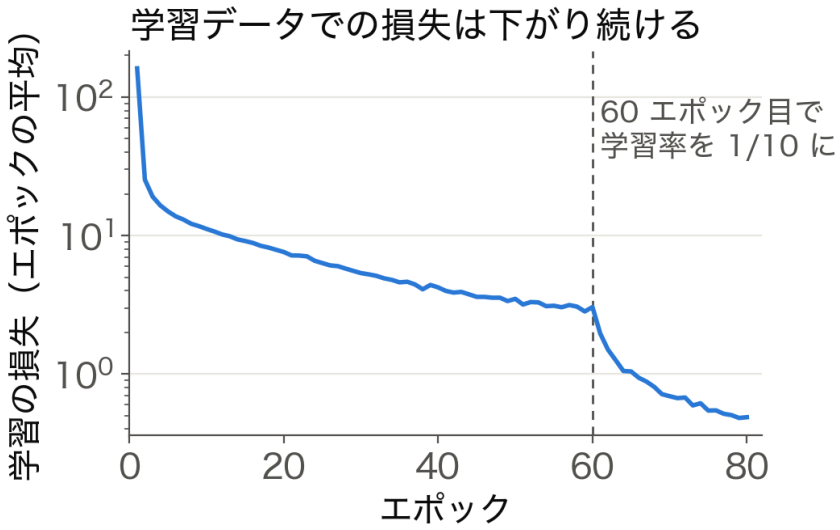
種類	AP50(1 回目)	AP50(続き)	AP50:95(続き)	試験の物の数
PETボトル	0.581	0.655	0.357	911
プラ容器	0.337	0.411	0.225	294
びん	0.441	0.546	0.211	86
缶	0.260	0.267	0.134	98
紙パック	0.217	0.291	0.125	162
平均(mAP)	0.367	0.434	0.211	1,551

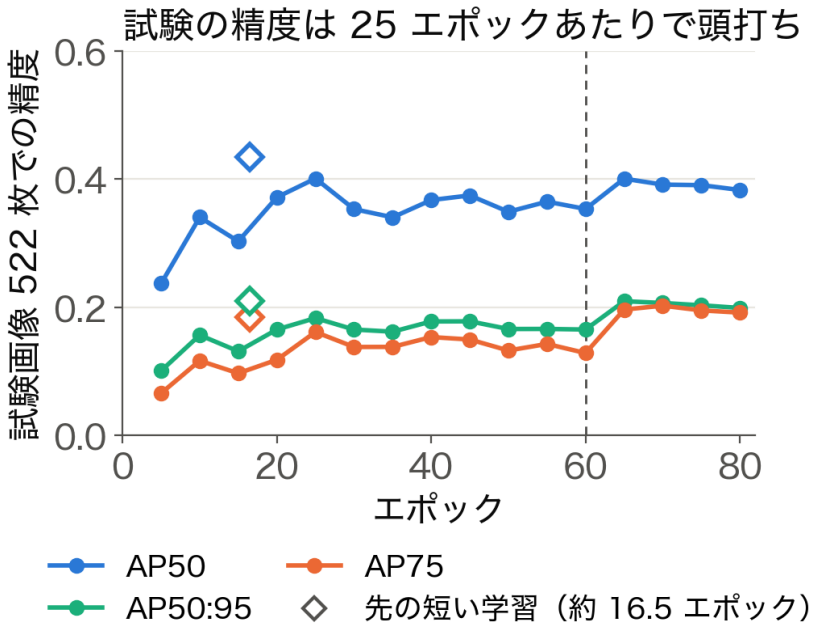
AP50: 枠が半分以上重なれば正解として数えた精度(1 が満点)。AP50:95 は重なりの基準を 0.5〜0.95 まで厳しくした平均。

上表より、学習を続けると缶以外は上がった。ただし、80 エポックまで延ばしても、これを超えなかった(p.22)。

分類の 88% と比べて低いのは、場所も当てる必要があり、重なりや隠れが効くためと考える。

検出(3)学習の長さ



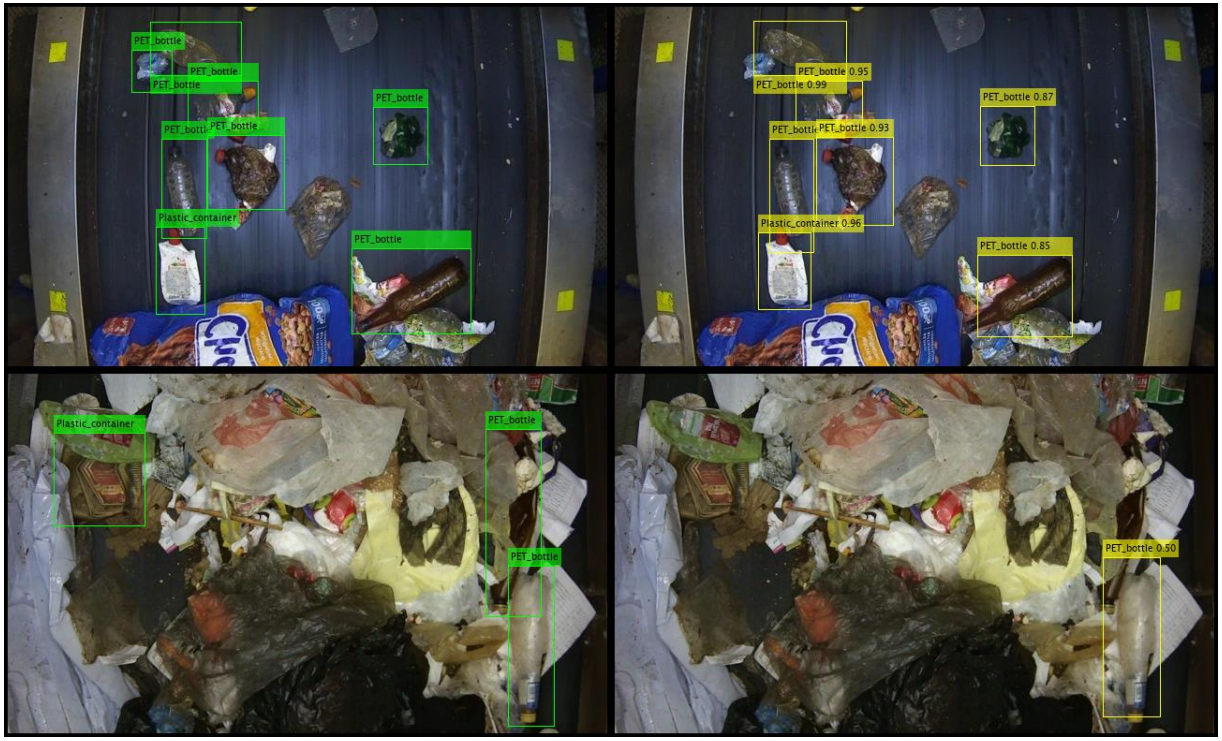


COCO で学習済みの物から 80 エポック学習し直し（学習率 1e-3、60 エポック目で 1e-4）、5 エポックごとに試験画像 522 枚で精度を測った（RTX 3050・146 分）。◇は p.21 の学習（約 16.5 エポック）。

上図より、学習データでの損失は下がり続けるが、試験の精度は 25 エポックあたりで頭打ちになる（AP50 0.40）。学習データに合わせすぎ（過学習）の形で、エポックを増やしても上がらない。

効いたのは学習率を下げたところで、60 エポック目に AP75 が 0.13 から 0.20 に上がった。精度を上げるには、水増しを強くする・解像度を上げる・大きい検出器にする方が見込みがあると考える（試していない）。

検出(4)例



左が人の付けた正解、右が AI の検出(続きの学習の後)。上はまばらな場面、下は山積みの場面。
上図より、まばらな場面ではよく見つけるが、山積みの場面では見落としが多い(下の例では 3 個のうち 1 個)。

（3）輪郭

検出した枠を SAM に渡して、物ごとの輪郭を切り出す

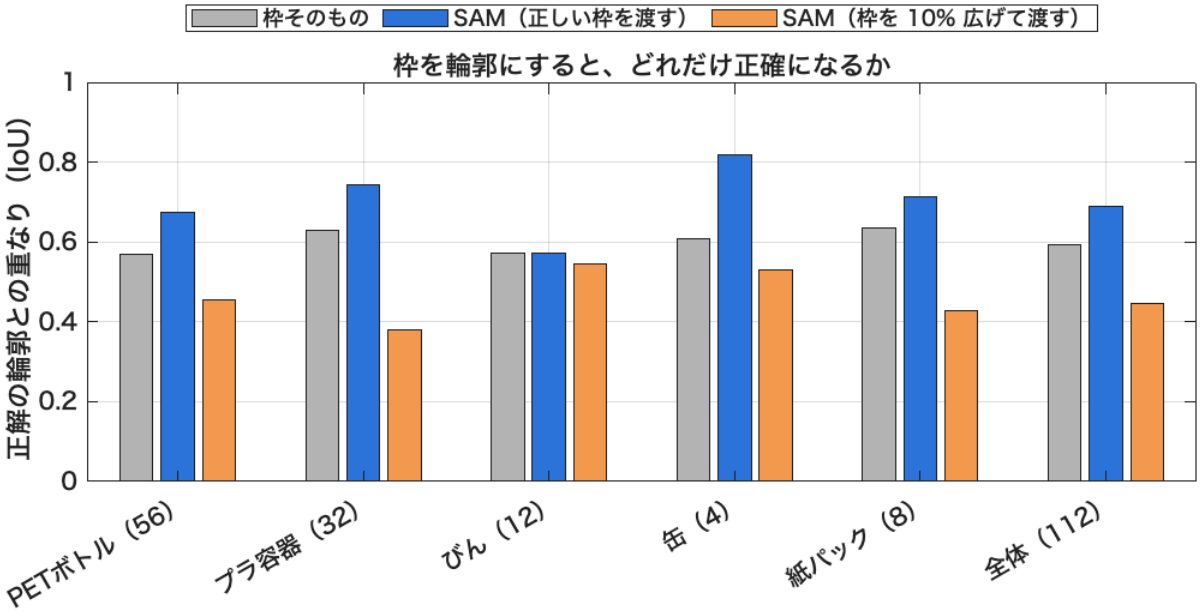
AI: SAM（学習はしない）

輪郭(1)方針

項目	内容
狙い	検出の枠(物を囲む長方形＝外枠)を、物の輪郭にする。面積の割合を出す・ロボットがつかむ位置を決める、には輪郭が要る。
方法	SAM(Segment Anything Model)に画像と枠を渡し、枠の中の物の輪郭を出させる。学習はしない。種類は検出器、輪郭は SAM が受け持つ。
道具	MATLAB R2025b の segmentAnythingModel(MathWorks 提供のアドオン)。画像ごとに特徴を 1 回計算し、枠の数だけ輪郭を出す。
正確さの測り方	WaRP-S(輪郭の正解付きの切り出し画像 112 枚)で、(1) 枠そのもの (2) SAM(正しい枠)(3) SAM(枠を 10% 広げる)の重なり(IoU)を比べる。
見本	試験画像のうち、検出のしきい値 0.3 で物が 3 個以上見つかった場面に輪郭を重ねる(図と動画)。

(3) は、枠を上下左右に 10% ずつ広げたもの。検出器の枠が正解より少し大きくずれた場合を想定した。

輪郭(2)結果

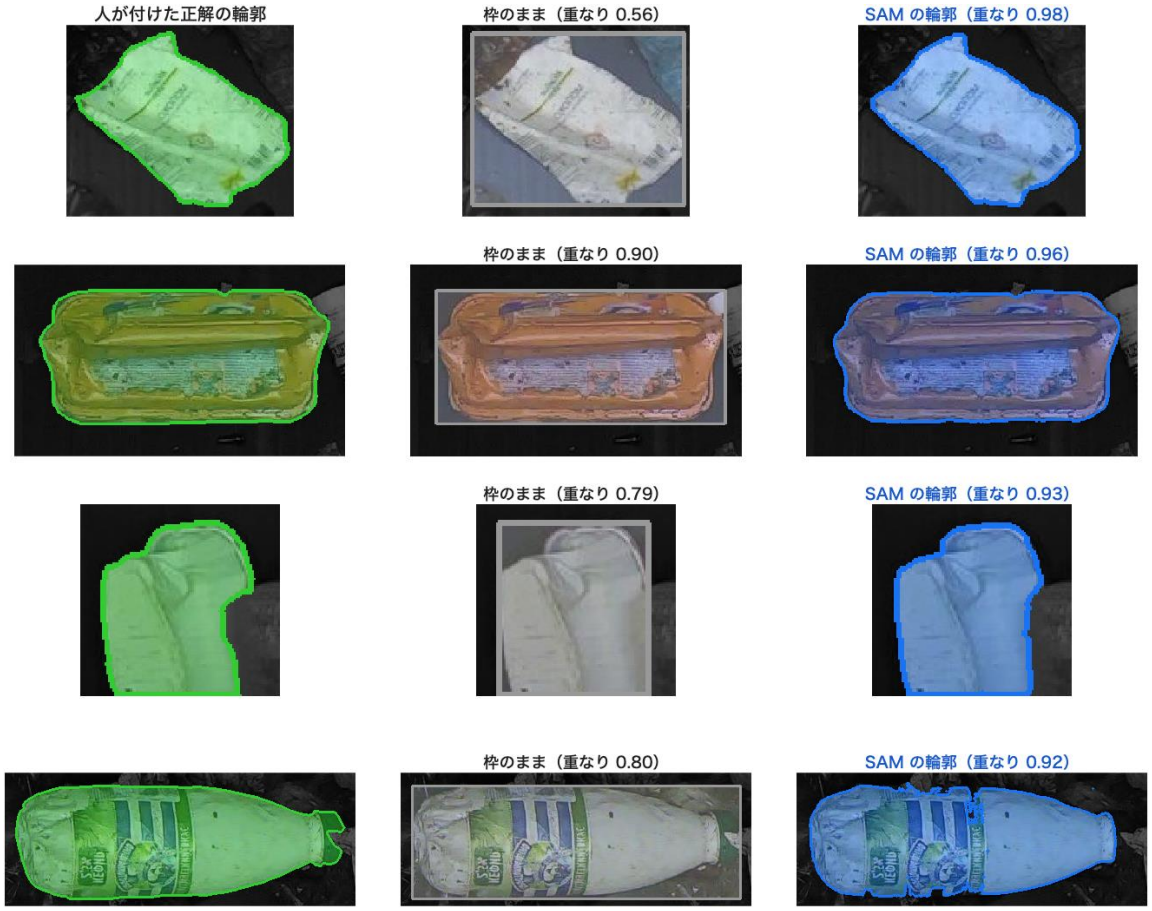


全体(112 枚)の IoU は、枠そのもの 0.592、SAM 0.690、枠を広げた SAM 0.445。SAM は 112 枚中 89 枚で枠を上回った。

上図より、SAM の輪郭は枠より物の形に沿う。枠を広げて渡すと隣の物まで含めるので、SAM の出来は渡す枠の正確さで決まる。検出を良くすれば輪郭も良くなる。

びんは SAM でも良くならない(0.571→0.573)。透明・暗い色のガラスは背景との境目が見えにくいためと考える。缶(4 枚)・紙パック(8 枚)は参考値。

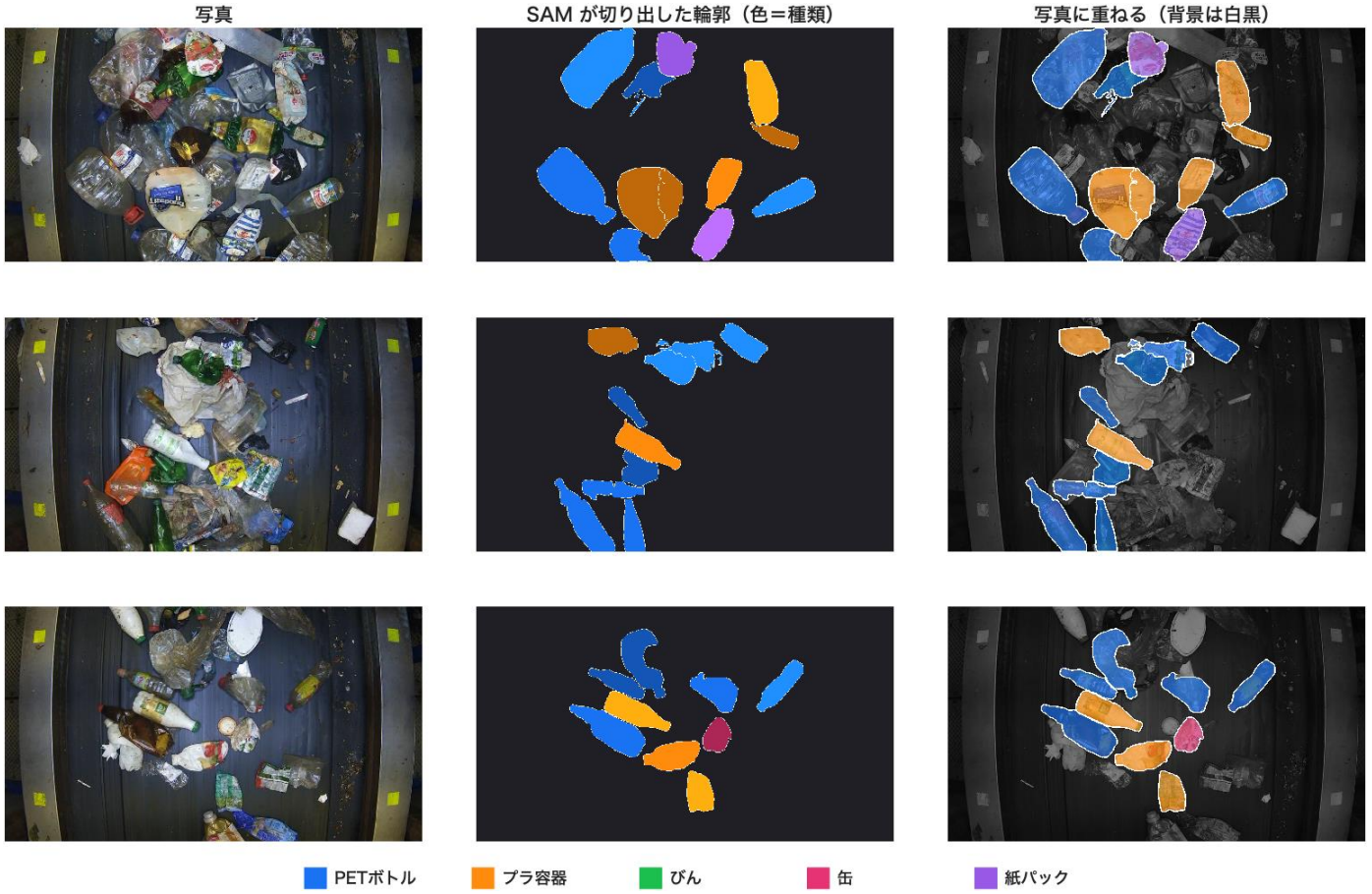
輪郭(3)正解との比較



左から、人が付けた正解の輪郭・枠のまま・SAM の輪郭。背景を白黒にして、物の部分だけに色を付けた（重なり＝IoU）。

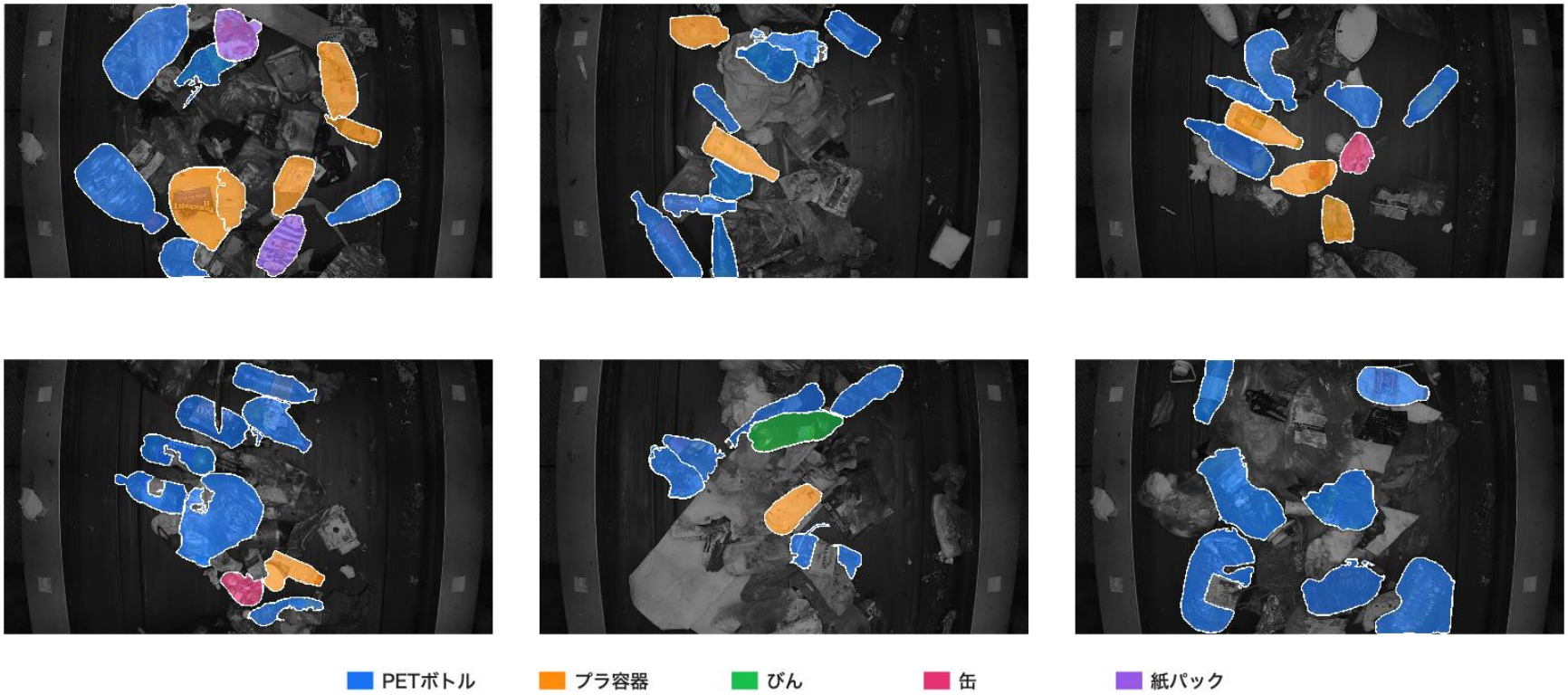
SAM がよく合った例から選んだ。112 枚の平均は、枠のまま 0.59、SAM 0.69（p.26）。

輪郭(4)物ごとの輪郭



左から、写真・SAM が切り出した輪郭だけ（色＝種類）・写真に重ねたもの。物の多い 3 場面。

輪郭(5)いろいろな場面



検出の枠を SAM に渡して、物ごとの輪郭を種類の色で重ねた(背景は白黒)。ノート PC の GPU でも 1 枚あたり数秒で輪郭が出る。

SAM 2 など、より速く動くように改良された新しいモデルも出てきており、さらに手軽に使えるようになっていく見込みである。

画像だけでは分からないこと

センサー	見分けられるもの	苦手なもの
カラーカメラ（今回）	形・色・見た目の材質、位置。	樹脂の種類（PET と PP など）、黒い物、中身。
距離も測るカメラ（3D）	高さ・重なり・つかむ位置。	材質。
近赤外の分光カメラ	樹脂の種類。	黒いプラスチック（近赤外を吸収する）。
中赤外のカメラ	黒いプラスチック。	装置が高い（研究段階）。
X 線	密度・厚み・元素、袋の中の電池。	細かい形。

参考: 弊社の調査報告（2026-10-21）の文献調査から要約した。

今回の「PETボトル」は、ボトルの形と見た目からの推定である。何を見分けたいかで、センサーを先に選ぶ必要がある。

まとめと次の手

- 現場向けの教訓は、次の 4 つである。
  - カメラの画素数より、照明とぶれ（シャッター速度）を整える方が効く。
  - 「全体で何%」ではなく、種類ごとの値で判断する。
  - 学習に使った日とは別の日の画像で、必ず確かめる。
  - 輪郭（SAM）の出来は、渡す枠の正確さで決まる。まず検出を良くする。
- 次に試す予定のことは、次のとおりである。
  - 日本の家庭ごみ（ペットボトル・缶・びん・紙パック）を自分で撮った写真で試す。
  - 検出の精度を上げる（学習を 80 エポックに延ばしても上がらなかったので、水増し・解像度・検出器の大きさを変える）。
  - 学習なしで分類できる AI（SigLIP2 など）と比べる。
  - 現場の写真が数百枚あれば、その現場での落ち方を同じ手順で測れる。

データとソフトウェアの表示・参考文献

- データ: WaRP（Waste Recycling Plant Dataset）。Kaggle で CC0（パブリックドメイン）として公開されている物を使った。
  - 論文: Yudin, D. et al., "Hierarchical waste detection with weakly supervised segmentation in images from recycling plants", SSRN 4183424。
  - 引用元: https://www.kaggle.com/datasets/parohod/warp-waste-recycling-plant-dataset。
- ソフトウェア: MATLAB R2025b（Deep Learning Toolbox、Computer Vision Toolbox、Image Processing Toolbox）。
- 学習済みの AI: ResNet-18、YOLOv4-tiny（COCO）。どちらも MathWorks 提供の物。
- 本資料の数値は、技術紹介のための試作で、条件ごとに 1 回ずつ学習して測った値である。実際の現場での精度を保証するものではない。