

画像分類＋物体検知アプリ

きのこの山とたけのこの里の二値分類

背景：

画像認識の基本を学ぶため、データ収集・学習・評価・アプリ化・デプロイまでの一連の流れを実践

ResNet18（画像分類）

- 背景：単一物体の分類を通じて、画像認識の基礎プロセスを学ぶ
- 課題：背景が変わると精度が落ち、複数物体が写ると対応できない

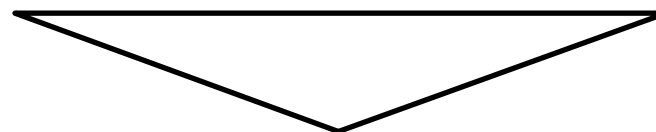
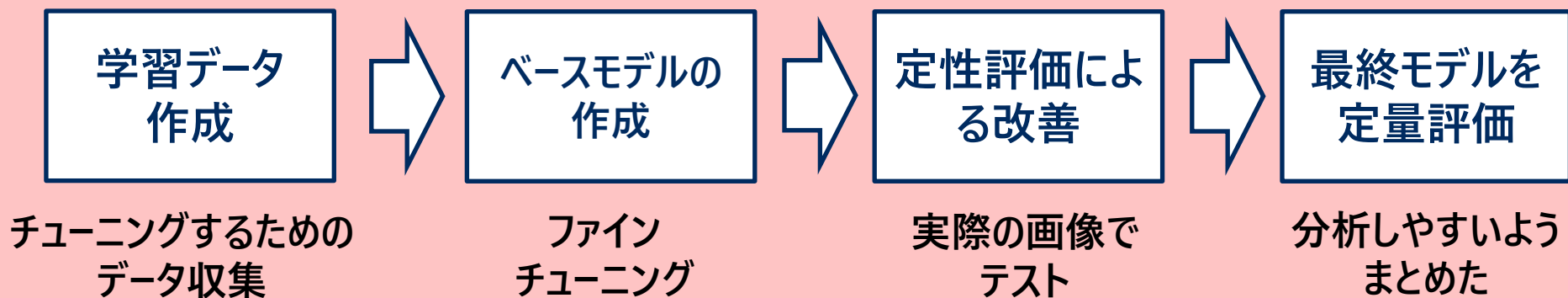
YOLO（物体検知）

- 背景：画像分類では限界があるため、複数物体やリアルタイム検出を解決する必要があった
- 課題：少量データでの汎化性能、ハイパパラメータ最適化、リアルタイム性の確保



きのこの山とたけのこの里の2値分類で実装（精度90%を目指す）

開発の流れ



デプロイ

画像分類アプリ

- ベースラインの作成
- 精度向上（汎化性能の向上）：画像の明るさ、角度などを変化させて学習 ⇒ 未知データへの対応力強化
- グレースケール化：色データを削除し、形状での分類
- 性能評価

表. 各モデルと性能指標

性能指標	Accuracy	Precision	Recall	F1-score
	全体でどれくらい正しくあてたか	ポジティブと予測したうちどれくらい正しかったか	本当にポジティブなものをどれくらい取りこぼさなかったか	PrecisionとRecallのバランス
値	1.00	1.00	1.00	1.00

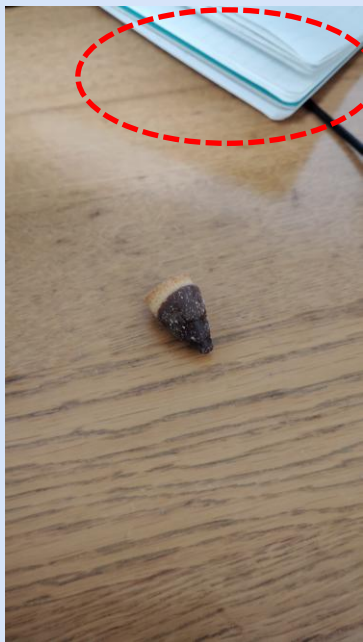
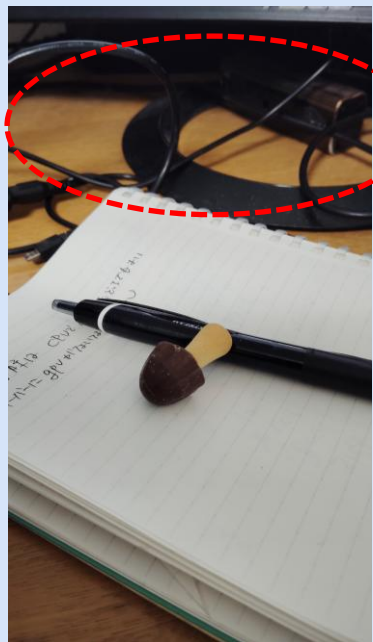
過学習による偶然の一致ではなくシンプルな画像であったため正答率が高かった



目標である精度90%を超えたのでデプロイ

画像分類アプリ（Resnet18）の限界

背景に複数の物体が含まれる場合



「きのこの山」や「たけのこ里」以外に、**他の物体（赤枠）**が含まれているため誤検出しやすい

きのこだけのこが複数ある場合



一つの画像に複数含まれていると「きのこの山」と認識してしまう

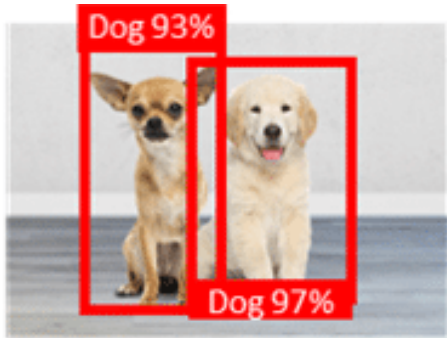


画像分類アプリの限界



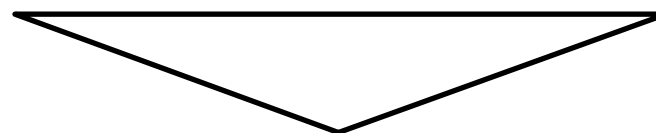
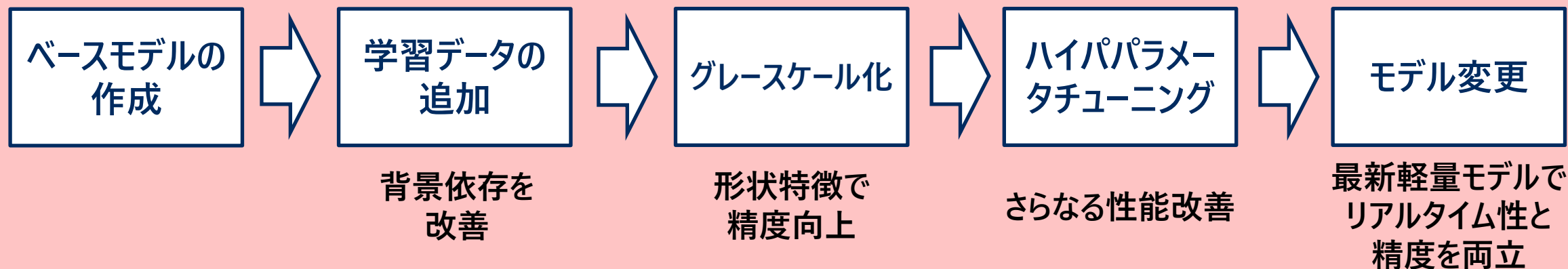
画像分類と物体検知（YOLO）の違い

背景に複数の物体が含まれる場合やきのことだけのこが同時に写っている場合に対応できるのは、物体検知ができるYOLO

項目	画像分類 (ResNet)	物体検知 (YOLO)
タスク内容	画像全体が何かを分類	画像内の物体を検出し、位置(bbox)とクラスを予測
例	「この画像は犬」	「この画像には左に犬、右にも犬がいる」
例（犬の画像）		
ラベル	クラスラベル（1枚=1ラベル）	bb ox座標 + クラスラベル（複数可）
アノテーション作業	簡易（分類タグ付けのみ）	bb oxアノテーションが必要（手間大）

YOLOを使って物体検知アプリを再度作成してみる

開発の流れ



デプロイ

- ・学習データの追加：背景依存を改善
- ・グレースケール化：グレースケール率を複数比較した結果、Gray=0.01が最良
- ・Gray=0.01でハイパラメータチューニング実施（計算リソースで1世代のみ）
- ・モデルの変更（YOLOv5→v8, v11、mAPで比較）

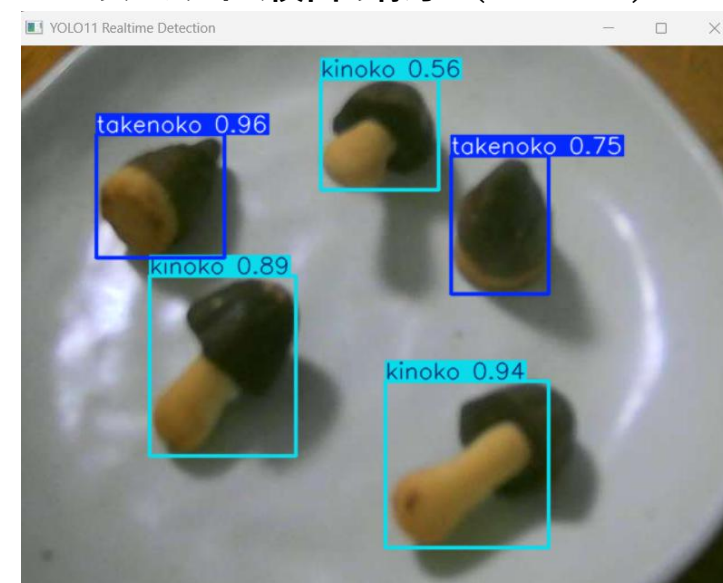
表. 各モデルと性能指標

性能指標\ 各モデル	YOLOv5 (チューニングあり)	YOLOv8 (チューニングなし)	YOLOv11 (チューニングなし)
Precision	0.979	0.981	0.991
Recall	1.000	1.000	0.981
mAP0.95	0.885	0.850	0.946

※各性能指標の値は最後のエポックでの値

目標であるmAP0.95：90%を超えたので
YOLOv11をベストモデルで決定し、デプロイ済み

リアルタイム検出の様子（YOLOv11）



画像分類

- ・画像データを追加して学習

物体検知

- ・YOLOv5でのハイパラメータチューニングのさらなる探索
- ・YOLOv11でのハイパーパラメータチューニングでの精度向上
- ・FAST API によるAPI化を検討
- ・マジックレイサー機能の追加