

LibreYOLOによる商用レベル物体検出モデルの構築

ライセンスの壁を超えたスケーラブルなAIパイプライン設計と実装プロセス

2026年10月17年

株式会社礎デザインオートメーション

商用化を阻む「ライセンスの壁」

YOLOファミリーの進化により、リアルタイム物体検出は飛躍的に進歩しました。しかし、最新モデルの多くはAGPLライセンスを採用しています。

ビジネス上の課題: AGPLの制約により、SaaSプラットフォーム、クローズドソースの商用プロダクト、または組み込みシステムへの統合には、高額なエンタープライズライセンスの購入が必須となります。

結論: 技術的に優れていても、ライセンス要件が商用展開の致命的なボトルネックになります。



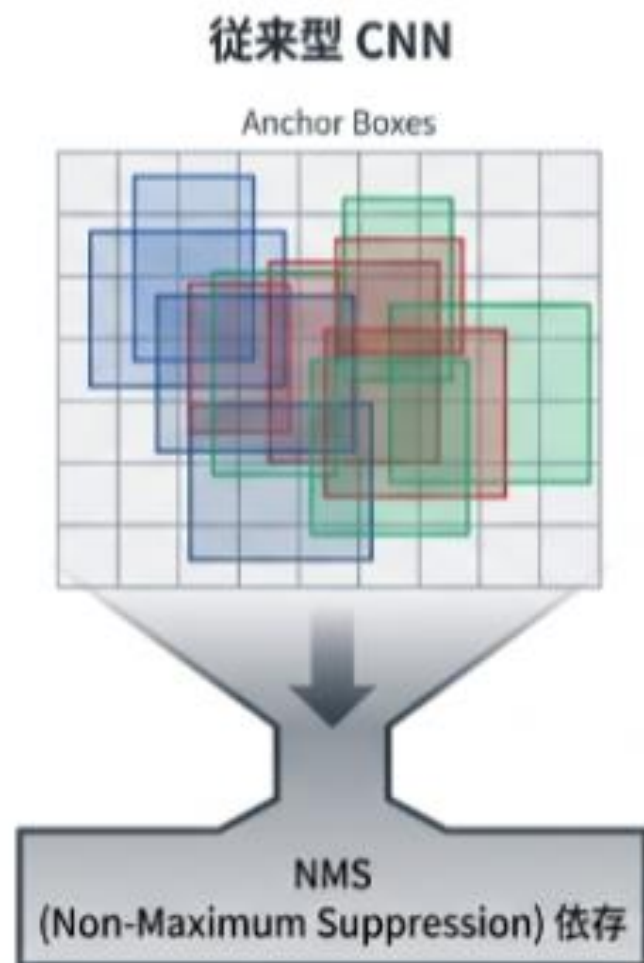
「LibreYOLO」によるブレイクスルー(MITライセンス)

評価項目	従来型YOLO	LibreYOLO
ライセンス	AGPL (制限あり)	MIT (完全オープンソース)
商用利用 (SaaS/クローズドソース)	エンタープライズ契約必須	✓ 安全・無制限に利用可能
対応アーキテクチャ	特定モデルに依存	✓ YOLOv9, YOLOX, RF-DETR等を単一APIで統合
デプロイメント	別途パイプライン構築	✓ 1行のコードでONNX, TensorRTへエクスポート

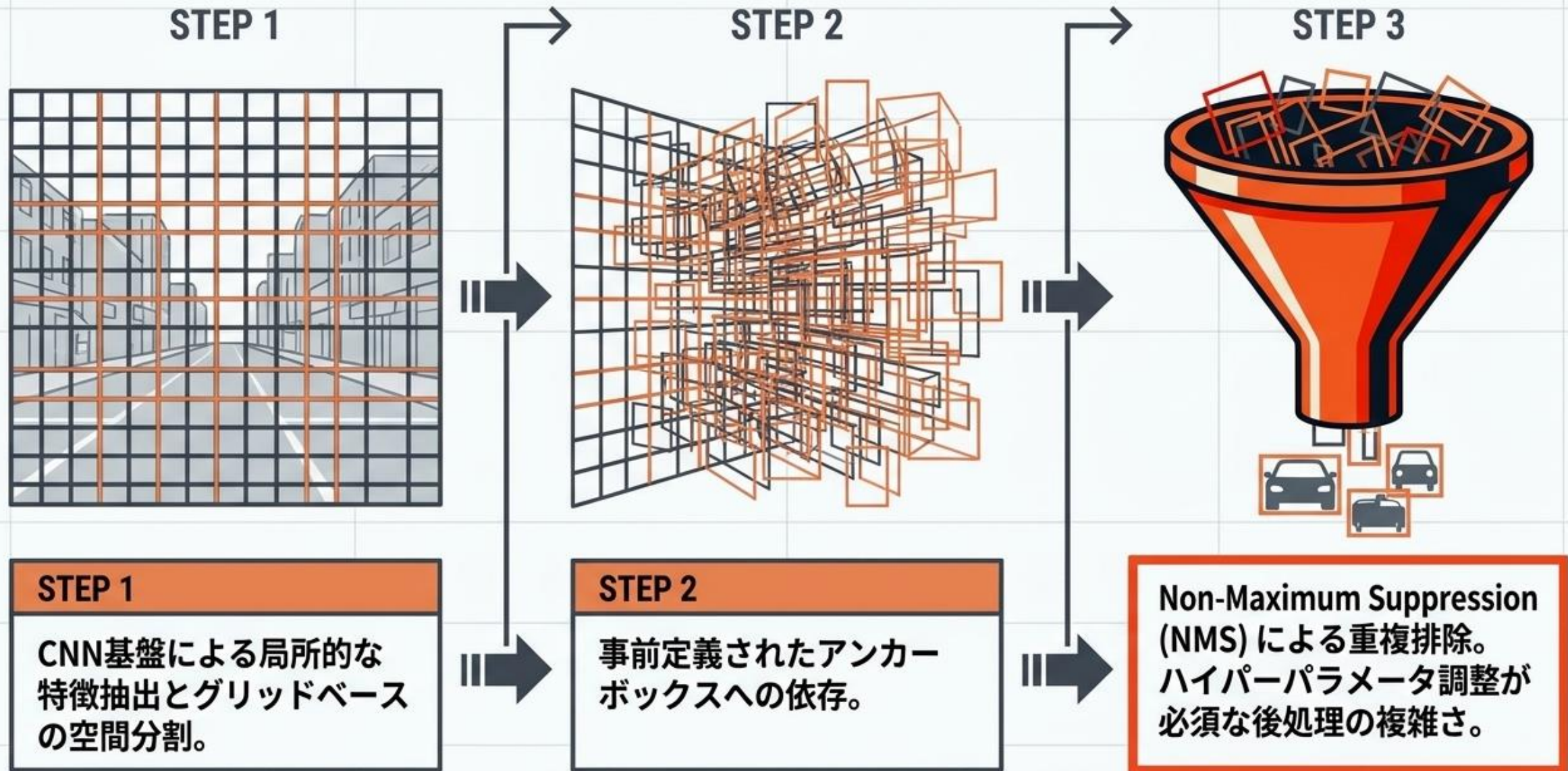
依存関係にAGPLコードを一切含まないため、商用展開とライセンス戦略を完全にコントロール可能。

なぜ、このワークフローでは、RF-DETRを選定するのか？

LibreYOLO提供の多様なモデルから、次世代アーキテクチャ「RF-DETR」を採用。

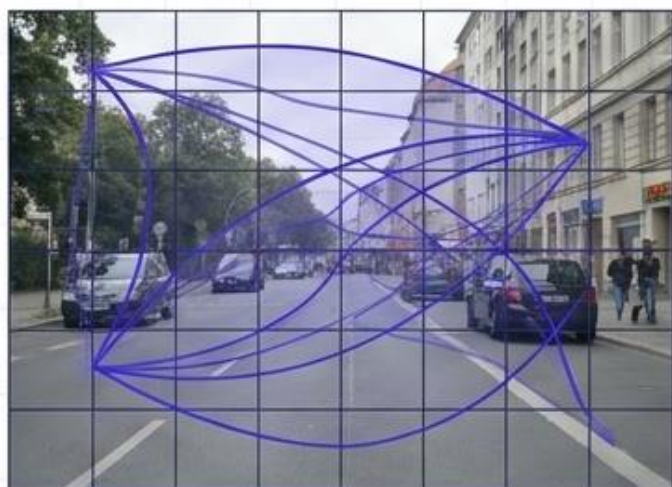


YOLOのメカニズム：局所的・ヒューリスティック



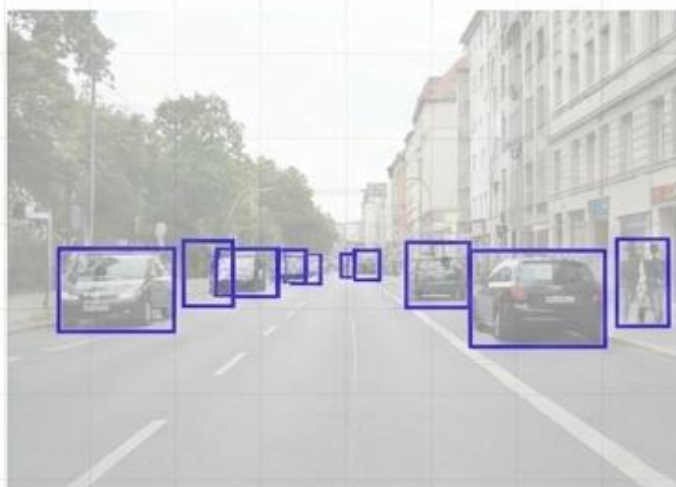
RF-DETRのメカニズム：大域的エンドツーエンド

STEP 1



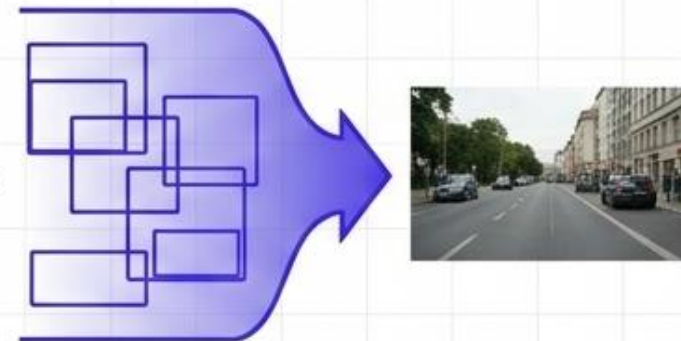
Vision Transformer基盤。
グローバルな受容野による
大域的な文脈理解。

STEP 2



Bipartite Matchingによる
セット予測。アンカーボックス
への依存からの脱却。

STEP 3



後処理（NMS）の完全排除。
エンドツーエンドの合理的な
アーキテクチャ。

YOLOが優位性のある領域

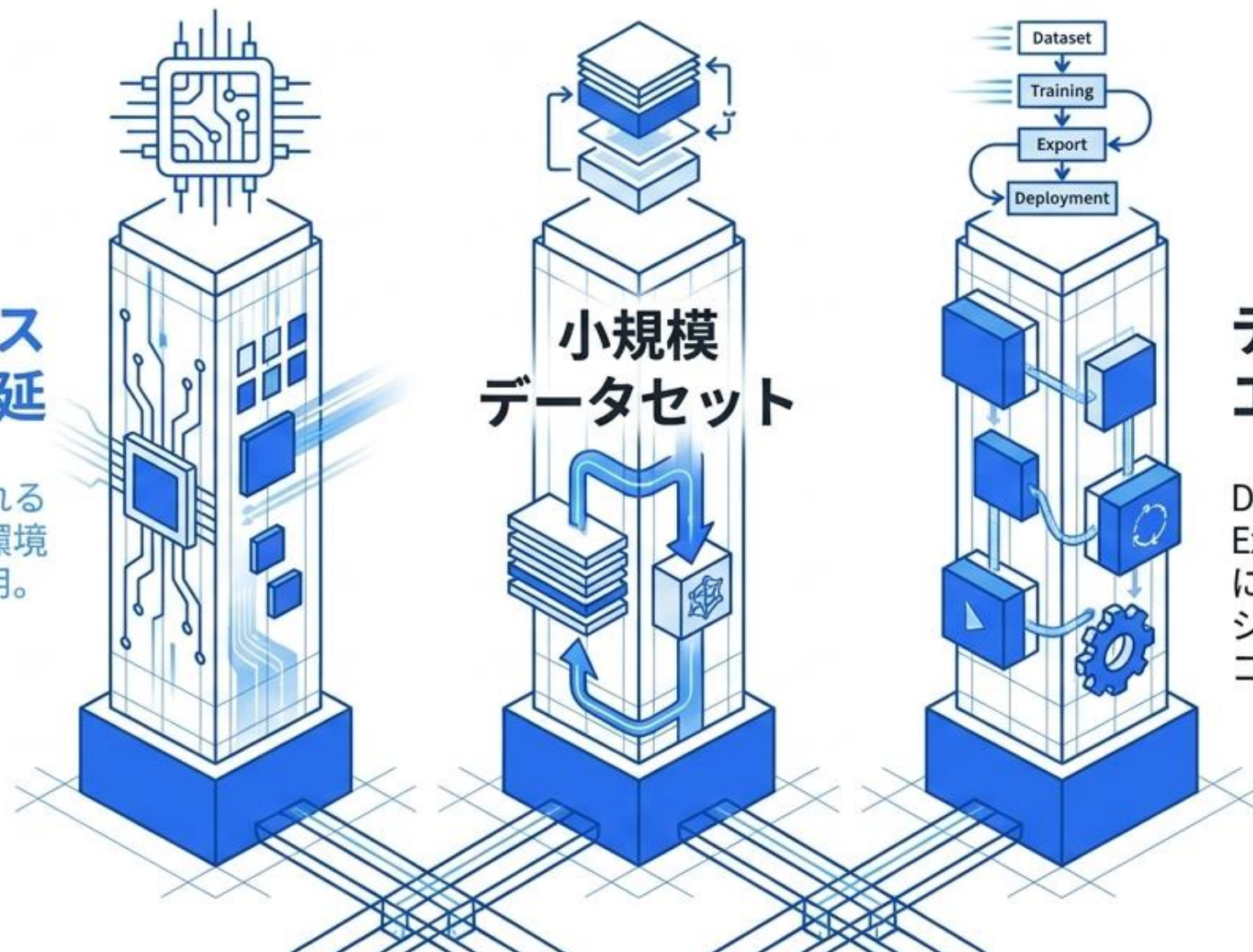
エッジデバイス & 超低遅延

数ミリ秒が問われる
モバイルやエッジ環境
での運用。

小規模 データセット

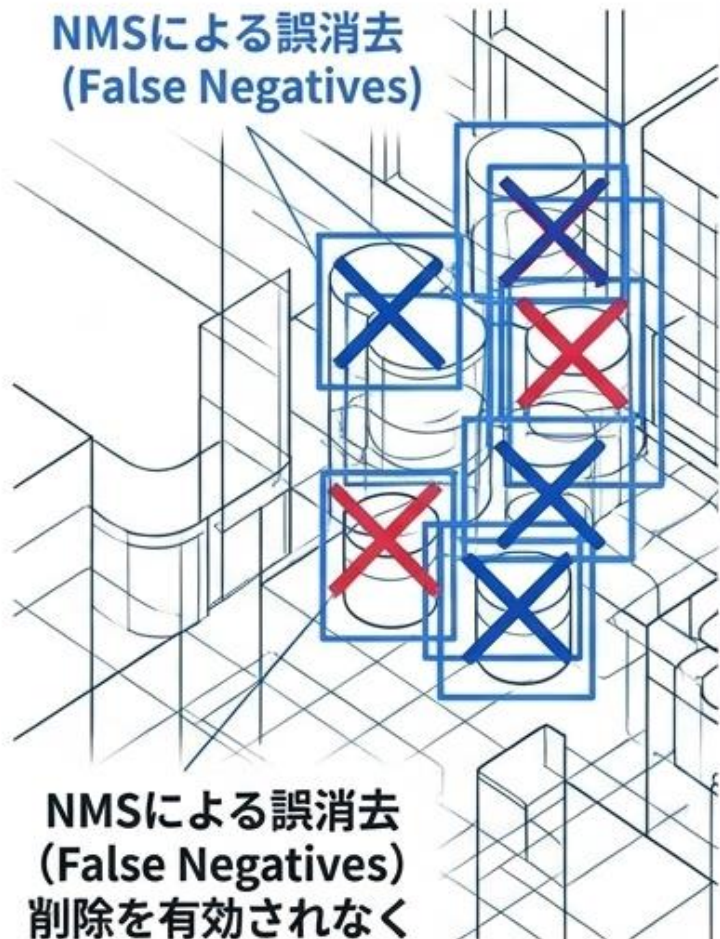
デプロイメント・ エコシステム

Dataset → Training →
Export → Deployment
における、圧倒的に
シームレスなツール群と
コミュニティの支援。

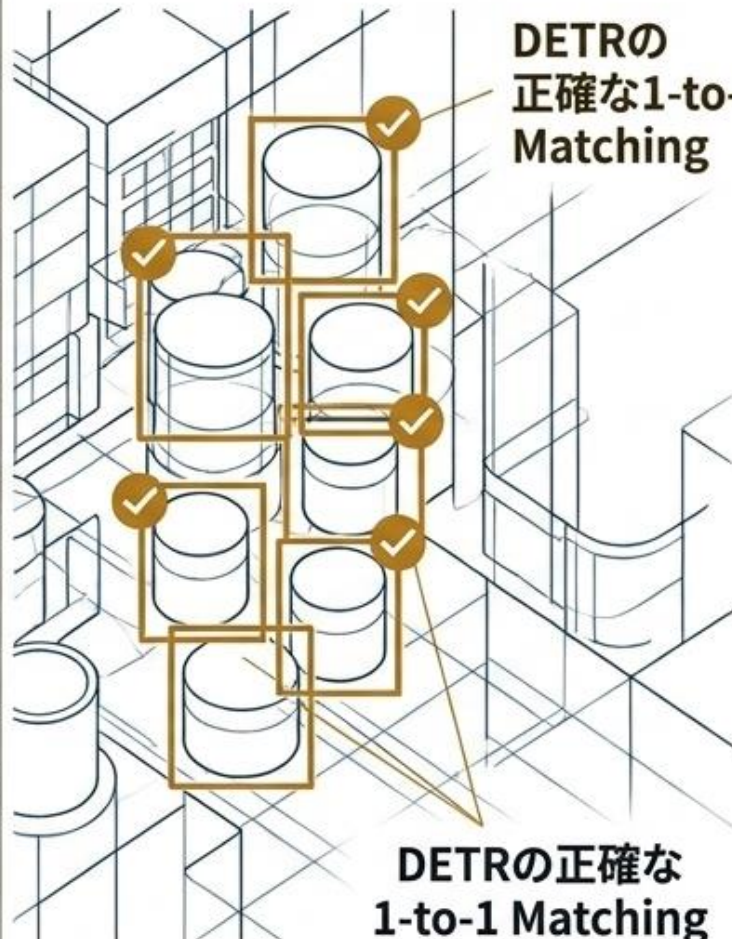


DETRの真価が発揮される複雑な現実世界

NMSによる誤消去
(False Negatives)



DETRの
正確な1-to-1
Matching



重なるの激しい密集環境

従来の検出器では、密集した予測が重なり合い、NMSが誤って正当なオブジェクトを消去するリスクがある。

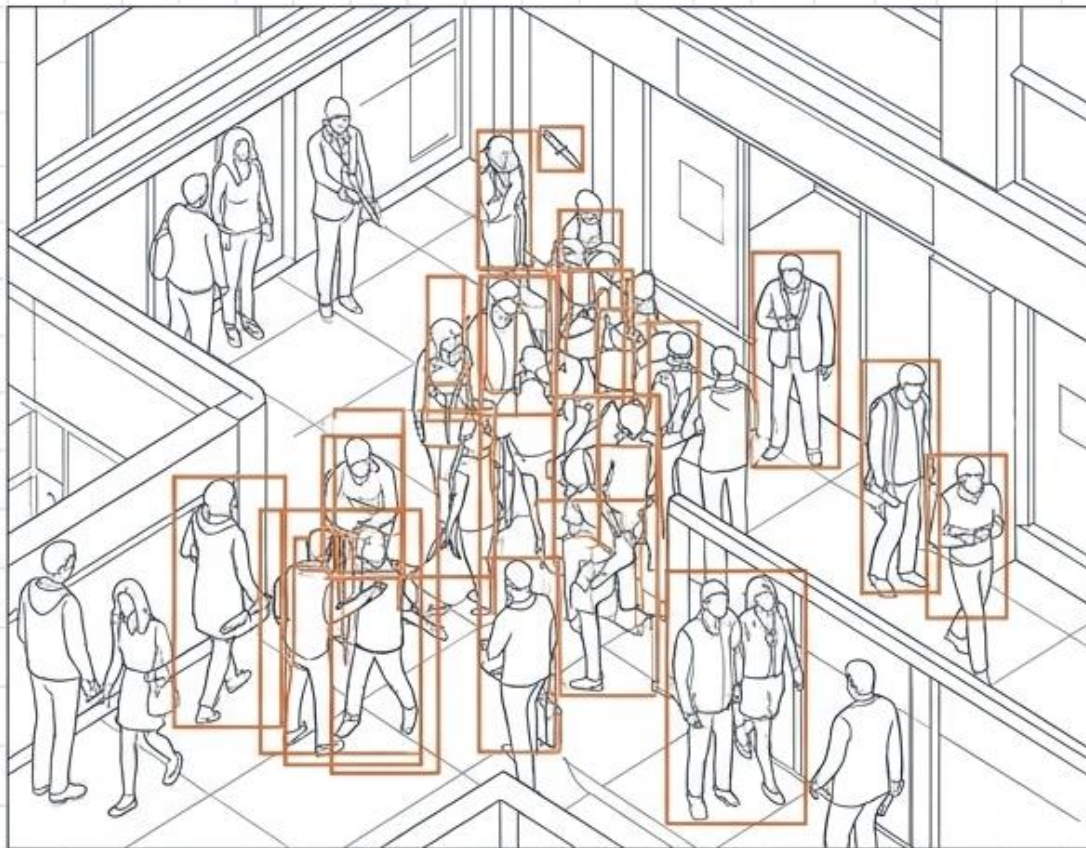
グローバルコンテキスト

TransformerのSelf-Attentionが画像全体の文脈（関係性）を捉えるため、重なりや部分的な遮蔽（Occlusion）に対して極めて高い堅牢性を持つ。

アーキテクチャの比較：YOLO vs RF-DETR

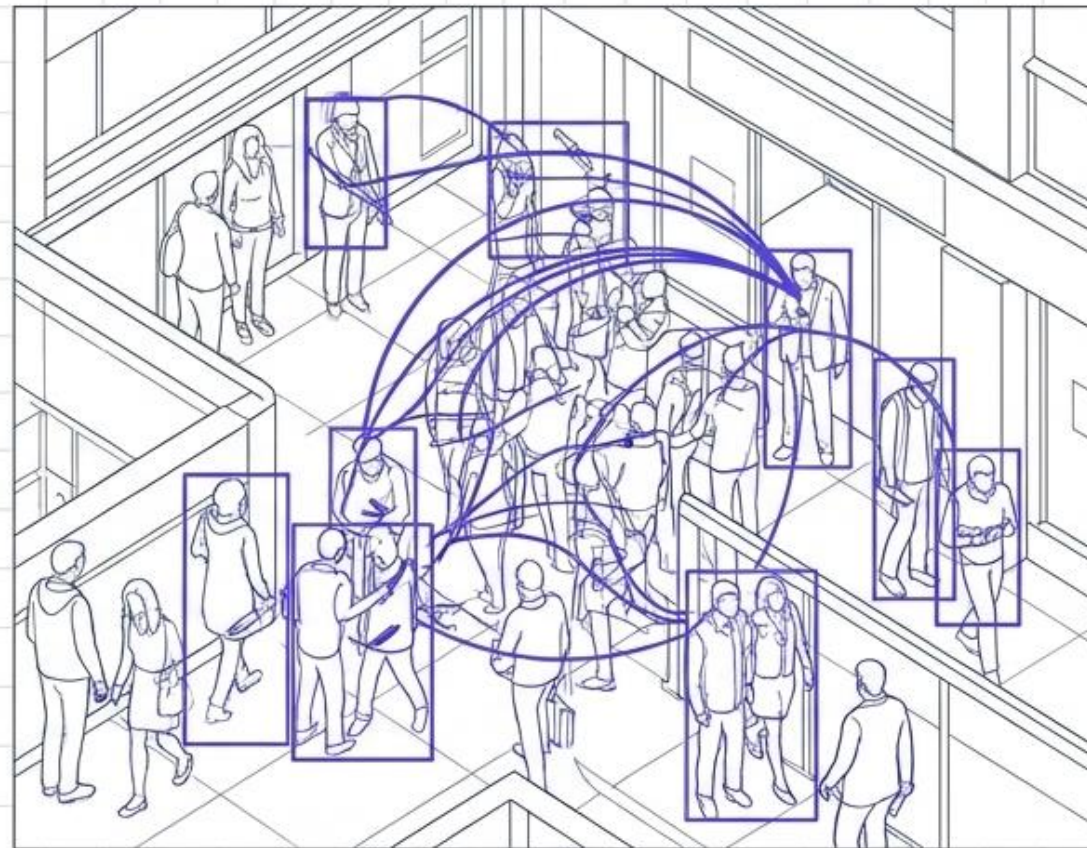
	YOLO	RF-DETR
空間理解	局所的（グリッドベース）	大域的（Global Attention）
複雑なシーン（密集・オクルージョン）	弱い（アンカーの限界による取りこぼし）	非常に強い（対象物間の関係性を認識）
パイプラインの複雑さ	高い（NMSやアンカーのチューニングが必要）	シンプル（NMSフリー、エンドツーエンド）
推論速度（リアルタイム性）	極めて高速（リアルタイム処理の基準）	実用的な高速化（NVIDIA L4で約20 FPS / 50msレイテンシを実現）

実世界でのパフォーマンス：なぜRF-DETRなのか？



YOLOの課題

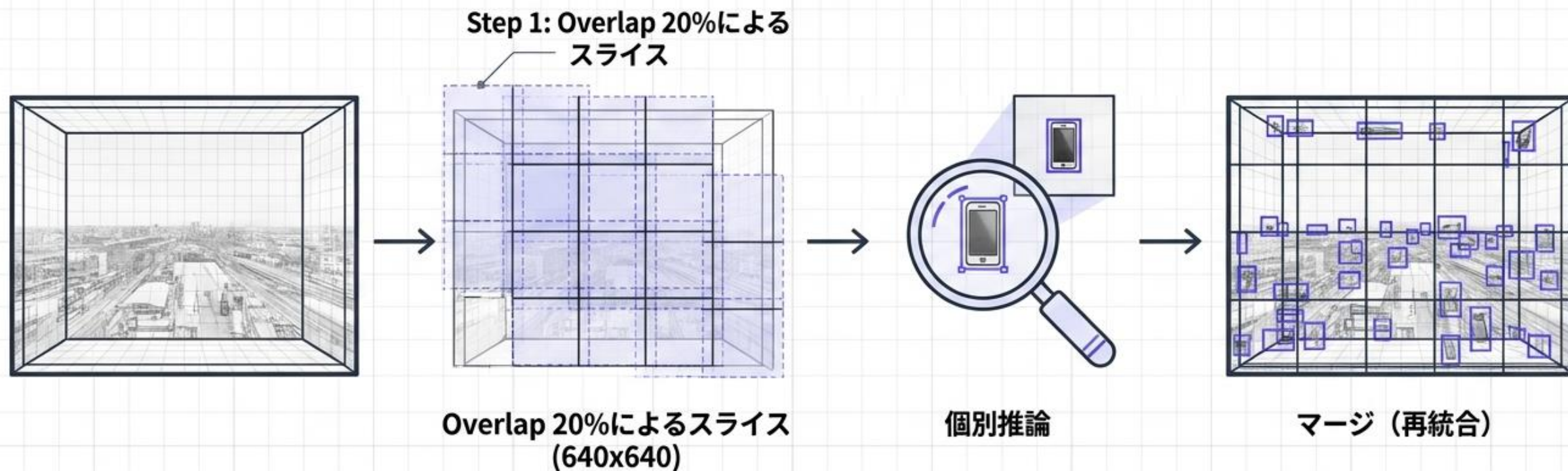
視覚的変化の激しい武器等の検出において、形状の欠落や重なりが致命的なエラーを引き起こす。



RF-DETRの優位性

複雑なセキュリティ脅威（多種多様な形状・角度）に対しても95%以上の精度を安定して達成。

補完技術としてSAHIによる微小物体検出の追加



ベースモデルを変更することなく、微小物体の検出能力を劇的に向上

コスト効率を極限まで高める学習インフラ設計

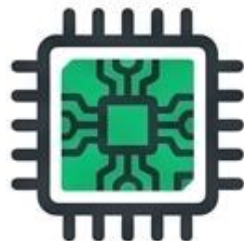
Cost-Savings Formula



GCP Spot VM × Custom Jobs

大幅な割引価格の
余剰リソースを活用

×



NVIDIA A100 (40GB)

大規模モデル学習に最適化

×



GCS (Google Cloud Storage)

高価な永続ディスクを排除し
直接ルーティング

×



FP16 混合精度学習

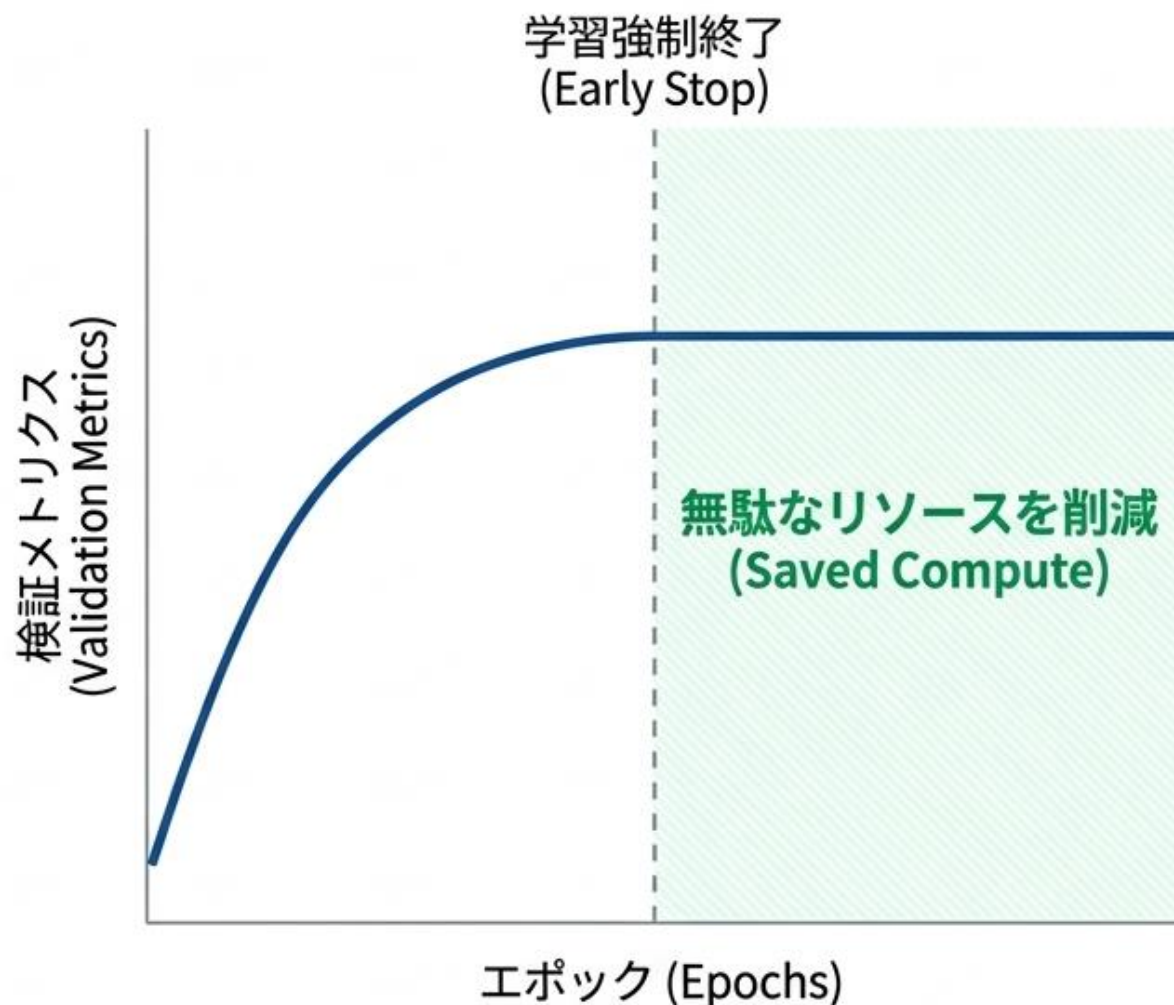
AMPサポートにより
GPUメモリ消費を半減



劇的な計算コストの削減とROIの最大化

リソースを保護する学習最適化：Early Stopping

- **厳格な検証モニタリング:** モデルのパフォーマンス（検証メトリクス）をエポックごとに監視。
- **早期終了の自動化:** パフォーマンスの向上が見られず、プラトー（平坦化）状態が20～30エポック連続した場合、学習プロセスを自動的に強制終了。
- **ビジネスインパクト:** 不要なGPU時間と莫大なクラウドインフラ費用を直接的に削減。Spot VMの強みと組み合わせることで、コストパフォーマンスを最大化。



マルチソース統合の泥臭いエンジニアリング(データの壁)

単一のパッケージ化されたデータセットに依存することは不可能。実運用に耐えうるデータを構築するための前処理パイプライン。



現実世界とデータセットの「アングルのギャップ」

解決策: 実世界のカメラ環境（エレベータッド・アングル）に特化したデータを意図的に収集・選定し、モデルが現実の視点に適応できるよう訓練。

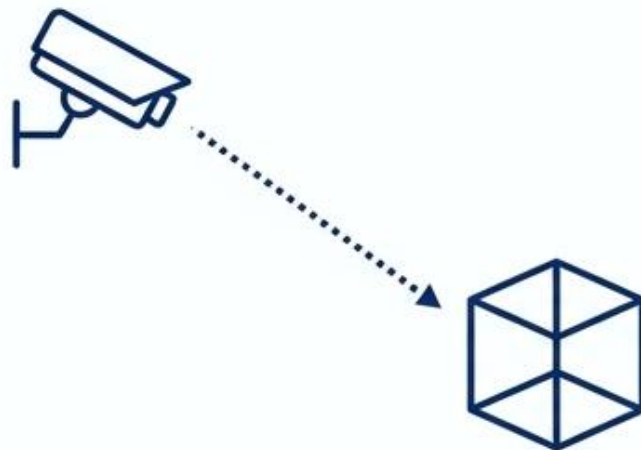
一般的な公開データセット



（標準的な視点・アイレベル）

- 公開データの多くは人間の目の高さで撮影されており、バイアスが存在する。

実際の監視カメラ環境

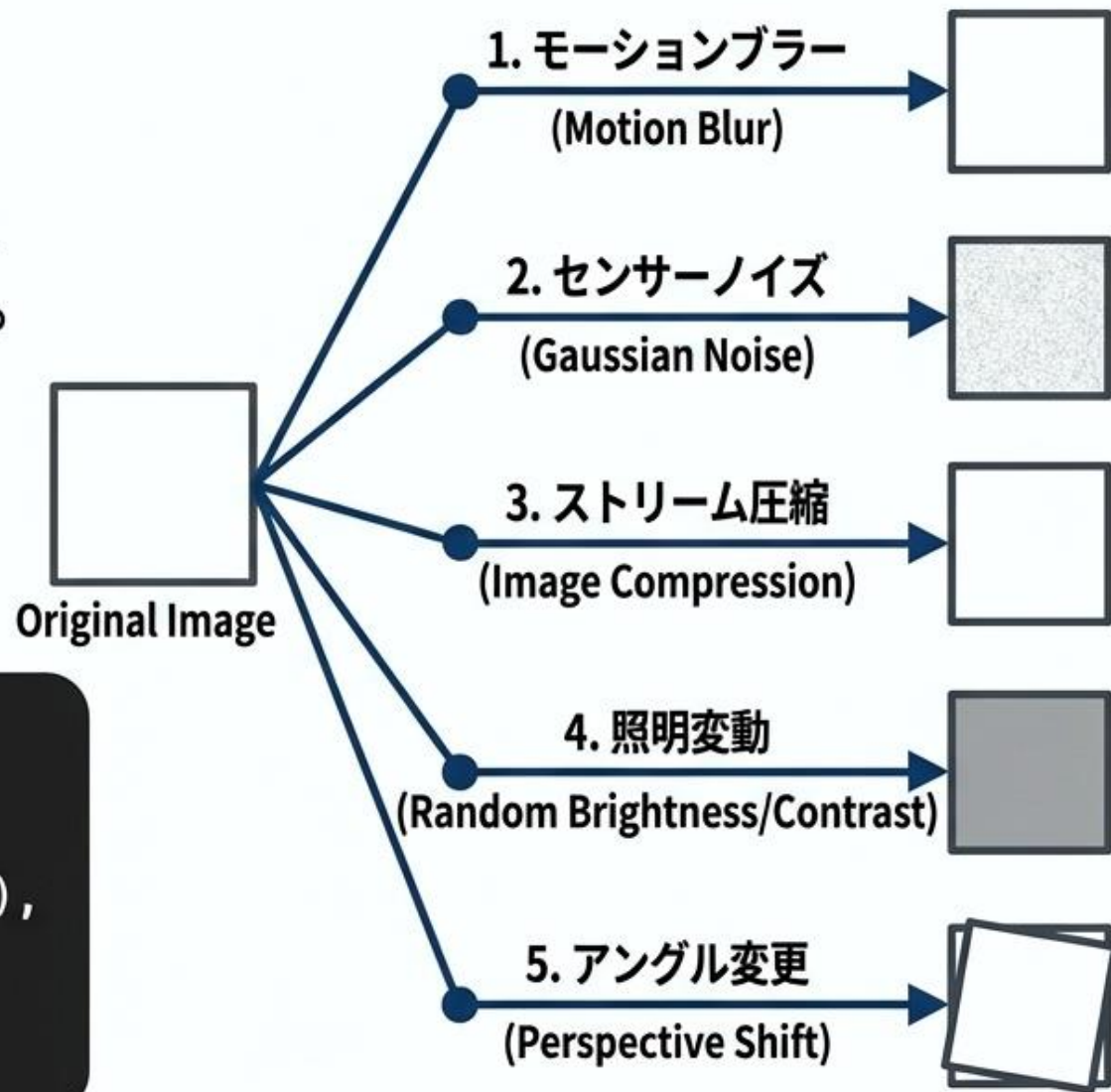


（見下ろし角・俯瞰）

- 高い位置からのアングルにより、物体の視覚的な特徴が大きく異なる。

堅牢性を生み出すデータパイプライン (Augmentation)

AI生成（合成）データはコストが高く現実感に欠けるため不採用。代わりに、パイプラインに直接強力なデータ拡張 (Albumentations) を組み込むことで、多様な環境に対する耐性を獲得。



```
augmentation = A.Compose([  
    A.GaussianBlur(blur_limit=(3, 7), p=0.3),  
    A.GaussNoise(var_limit=(10, 50), p=0.3),  
    A.ImageCompression(quality_lower=60, p=0.4),  
    A.RandomBrightnessContrast(p=0.5),  
    A.Perspective(scale=(0.02, 0.08), p=0.3),  
])
```

1)

本番運用レベルの検出精度 (mAP@50)

パイプラインは初期のベンチマークを大きく上回り、実世界でのデプロイ要件をクリア。

Metrics Dashboard

低分散オブジェクト (顔、ナンバープレート、スマホ等)



視覚的な変化 (バリエーション) が少ないオブジェクトにおいて、極めて高い精度を記録。

複雑なセキュリティ脅威 (武器、ナイフ、銃等)



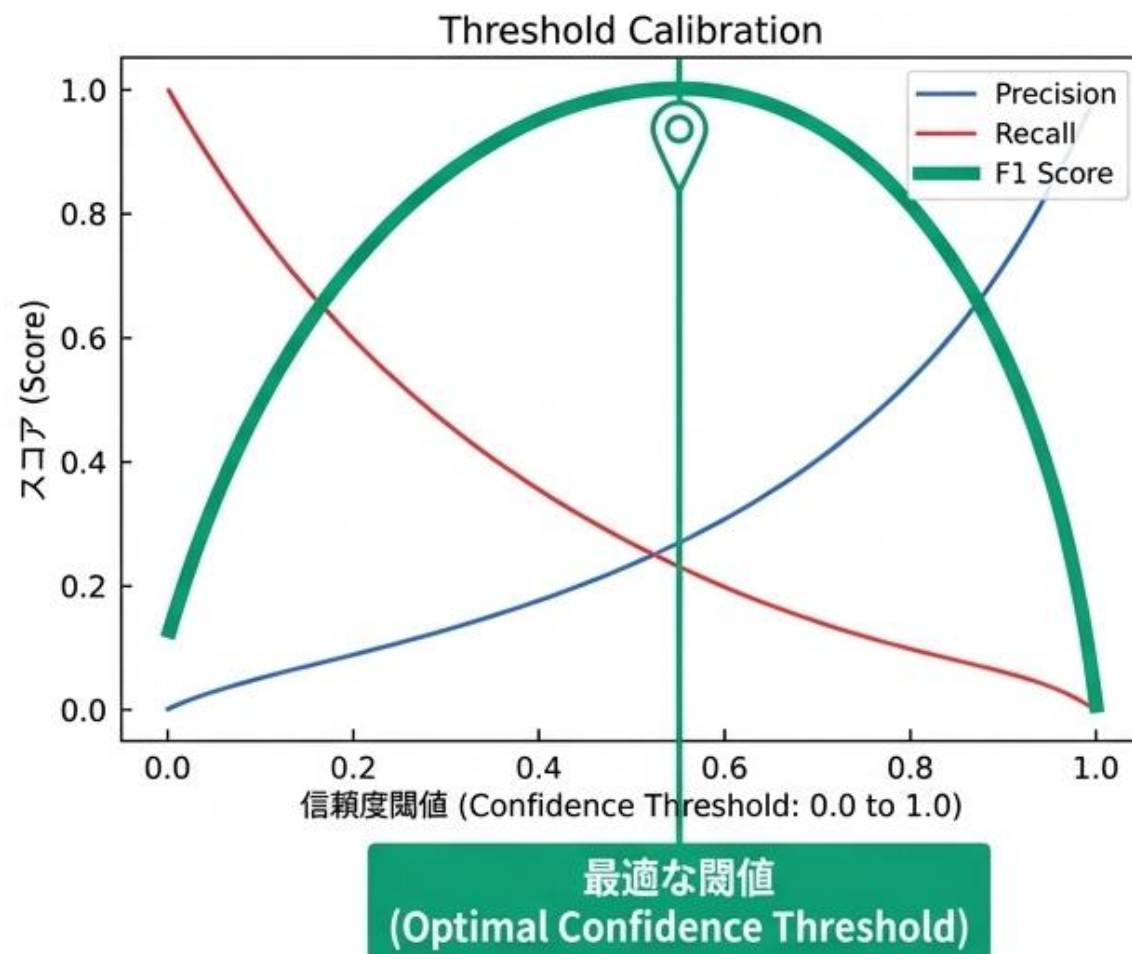
形状、サイズ、角度が多岐にわたる複雑な物体であっても、本番環境のセキュリティアラートとして十分な信頼性を確保。

F1スコアによるキャリブレーション(信頼度閾値の最適化)

静的で根拠のない閾値を設定することは、本番環境でのよくある失敗です。

- 自動スイープ: 0.01から0.96までの閾値に対して PrecisionとRecallを計算。
- F1ピークの特定: 誤検知の最小化と検出感度の最大化の数学的な最適解となる「F1スコアの頂点」を特定し、本番用の閾値として採用。

```
for conf in CONF_THRESHOLDS:
    results = model.val(data='dataset.yaml', conf=conf)
    p = results.results_dict['metrics/precision(B)']
    r = results.results_dict['metrics/recall(B)']
    f1 = 2 * p * r / (p + r + 1e-9)
```



リアルタイム推論パフォーマンス

どれほど高精度でも、ライブビデオの処理が遅ければ実用性はありません。NVIDIA L4 GPUを用いたベンチマークにおいて、リアルタイムCCTV ストリーム分析の厳しい要件を余裕でクリアするスピードを達成しました。

20 FPS

Frames Per Second



~50ms

Latency (レイテンシ)

本番運用に向けたエンジニアリングの学び

1. ライセンスの罣

「オープンソース」を謳うライブラリの多くがAGPLの罣を持つ。LibreYOLO (MIT) の発見が商用化への最大ブレイクスルーとなった。

2. 依存関係の管理

環境差異によるサイレントな障害を防ぐため、完全な依存関係をパッケージした独自の「ゴールデンイメージ」の構築が必須。

3. データ収集のボトルネック

CCTV特有のアングルに適合する高品質な商業用データを見つけ、クレンジング・結合するプロセスが、最も時間と労力を要した。

4. ラボと現実のギャップ

テストデータでの高精度はライブ映像での成功を保証しない。ハードサンプルの収集と再学習の継続的なループが不可欠。

総括: 技術的なアーキテクチャだけでなく、インフラコスト、データ、ライセンスを総合的に設計することが、商用AIパイプライン成功の鍵となる。