
Mech-DLK

Mech-Mind

2022年05月12日

CONTENTS

1	ディープラーニングについて	4
2	ディープラーニングの実行方法と応用プロセス	7
3	ディープラーニングを代表的なプロジェクトに適用する	43
4	Mech-DLK 使用ガイド	48
5	FAQ	60

Mech-Mind 株式会社は、成熟期のビジョン製品とディープラーニングアルゴリズムの併用により、世界中の自動車、3C、製造、物流などの業界に効率的なトータルソリューションを提供しています。画像分類、欠陥検出、属性認識など、様々な **知能認識** の細分化されたニーズに対応しています。

クイックスタート

ディープラーニングについて

ディープラーニングの実行方法と応用プロセス



インスタンスセグメンテーション

画像内の対象物の輪郭、位置、カテゴリを認識



画像分類

物体の画像カテゴリを認識

ディープラーニングを代表的なプロジェクトに応用



紙箱のデパレタイジング・パレタイジングのプロジェクト

ビジョンガイドによるプロセス



麻袋のデパレタイジング・パレタイジングのプロジェクト

ビジョンガイドによるプロセス

Mech-DLK ソフトウェアの使用ガイドとディープラーニングの環境構成

Mech-DLK 使用ガイド

ディープラーニング・トレーニング・ソフトウェア

環境構成

ディープラーニングの環境構成とトラブルシューティング

よくある質問と回答

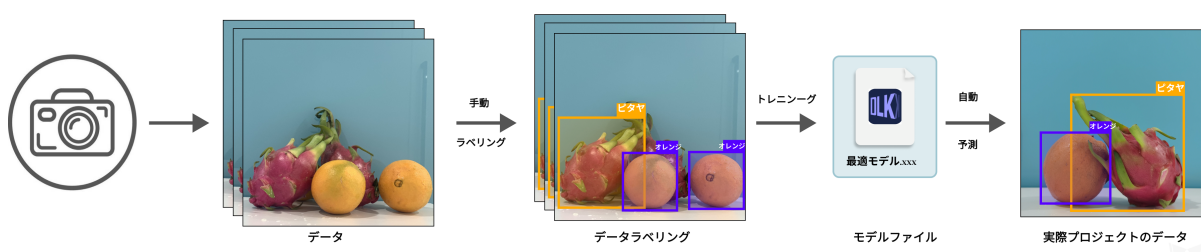
FAQ

よくある質問と回答

ディープラーニングについて

ディープラーニングとは、「データの特性を学習することで問題を解決する」ことを意味します。

認識する対象物の画像を撮影し（取得した画像は「データ」と呼ばれる）、画像上で認識する特性をマークし、（このプロセスは「ラベリング」と呼ばれる）、Mech-DLKで画像とラベリングを使用することで物体の特性を学習してルールを要約して（このプロセスは「トレーニング」と呼ばれる）、結果（ファイルで表示され、「モデル」と呼ばれる）を取得します。認識を実行するために、この結果を「データ」と類似した実際プロジェクトに応用（「予測」と呼ばれる）します。



ディープラーニングを通じて以下のことを実現できます。

Mech-Mind 株式会社のディープラーニング製品は様々な問題を解決するソリューションを提供しています。

	インスタンスセグメンテーション	画像分類
代表的なプロジェクト	紙箱、麻袋、通い箱のデパレタイジング・パレタイジングワークのロードとアンロード商品仕分け宅配便小包仕分け...	ロードとアンロード作業にワークの種類、サイズ、向きなどを認識組み立てと把持作業に物体が正しく配置されているかどうかを判断紙箱、麻袋のデパレタイジング・パレタイジング作業に箱の色を区別...
プロジェクトの特徴	位置決め分類	分類

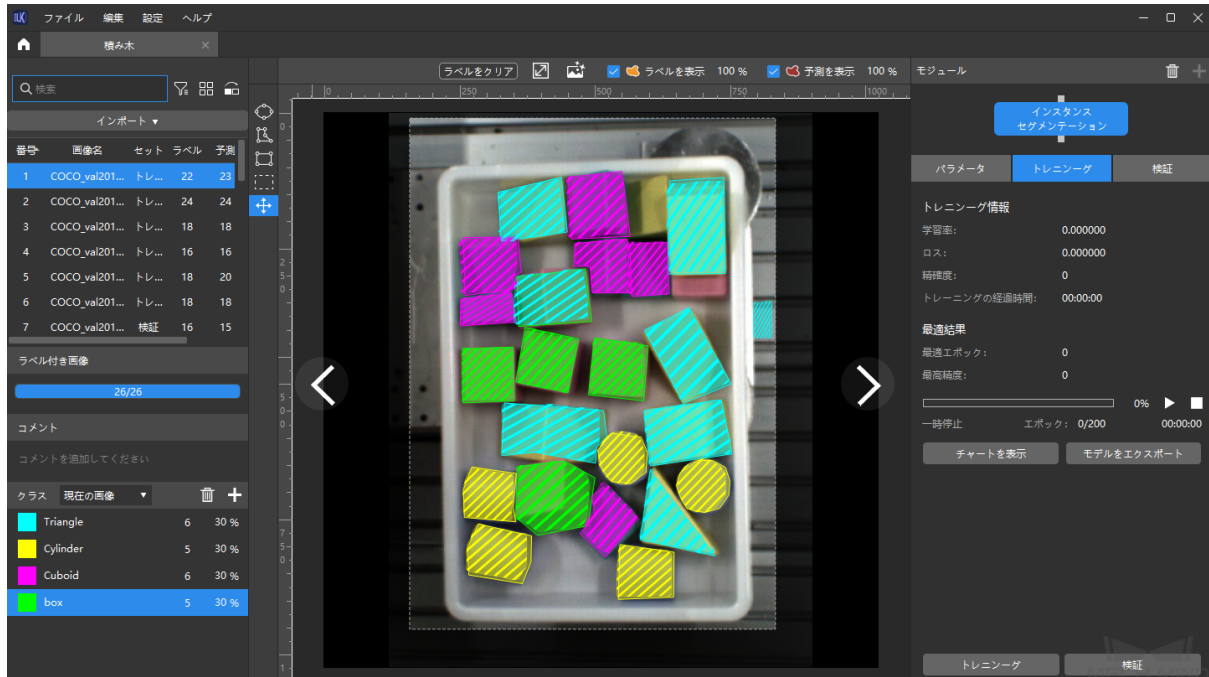
注意: ディープラーニングのアルゴリズムには、それぞれの強みがあり、互いに矛盾することはありません。プロジェクトの応用シーンと現場のニーズに応じて一つだけのアルゴリズムを使用するか、複数のアルゴリズムを使用してください。

ディープラーニングのプロセスは通常、以下5つのステップがあります。

ステップ1 環境インストール：詳しくは、[環境構成](#) をご参照ください。

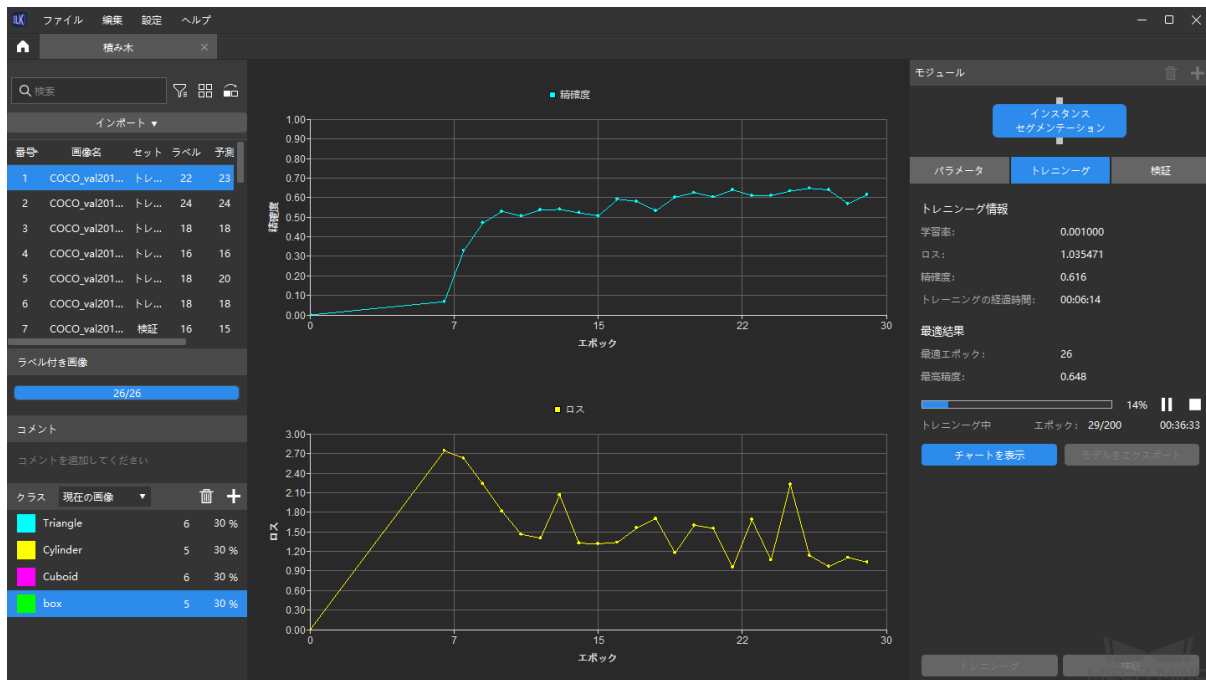
ステップ2 データ準備：データを取得、ラベリング、およびチェックします。

1. 実際の応用シーンと同様の条件でディープラーニングに必要な画像データを取得します。
2. プロジェクトの要件に応じてアルゴリズムを選択し、ルールに従ってデータセットをラベリングします。
3. ラベリングが誤ったデータがあるかどうかをチェックします (Mech-DLK を使用すればトレーニングに必要なデータセットを効率的に準備できます)。



Mech-DLK データラベリング

ステップ3 モデルトレーニング：ラベリングしたデータを使って、Mech-DLK で必要なディープラーニングモデルをトレーニングします。詳しくは、[クイックスタート](#) をご参照ください。



Mech-DLK ディープラーニングモデルをトレーニング

ステップ 4 効果評価：事前に準備したテストセットを使用してモデルの効果が要件を満たしているかどうかを評価します。

ステップ 5 予測：ユーザーは、評価されたモデルをプロジェクトの新しい画像に応用できます。

ディープラーニングの実行方法と応用プロセス

2.1 インスタンスセグメンテーション

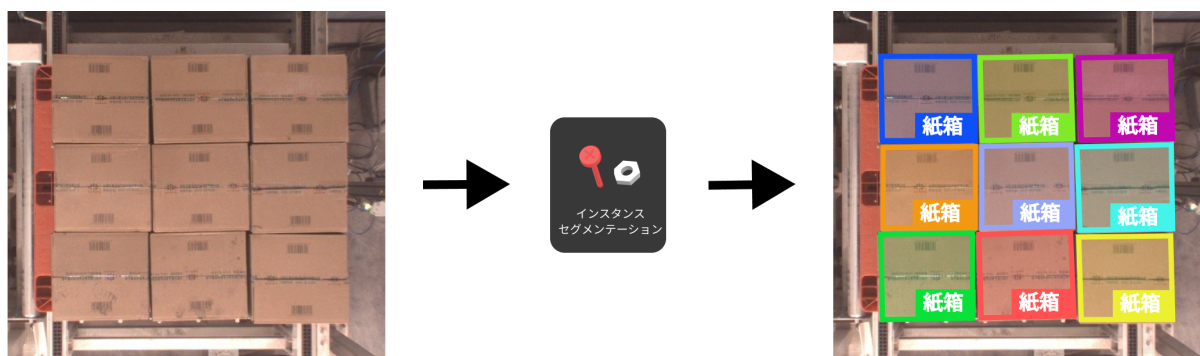
2.1.1 全体的な紹介

インスタンスセグメンテーションの機能

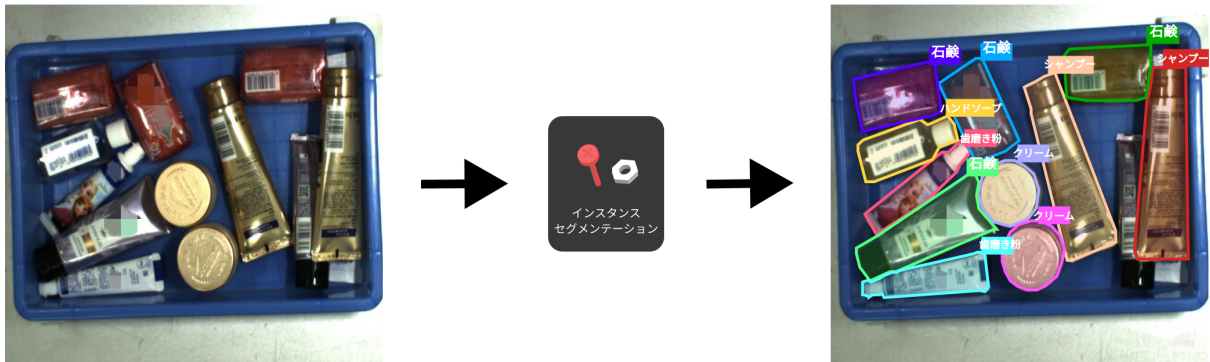
インスタンスセグメンテーションによって解決される問題は、「何がある、何が、どこに」です。即ち、画像に対象物があるかどうか、対象物の種類、画像にある位置を判断します。

具体例：

- 対象物が紙箱である場合、画像内に紙箱があるかどうかを判断します（モデルは紙箱以外の物体を認識しません）。ある場合は、各紙箱の輪郭をマークし、「紙箱」ラベルを表すカテゴリを出力します（ラベルは画像の特徴をマークしたときに作成されました）。紙箱がない場合は、結果は表示されません。



- 対象物が石鹸、歯ブラシ、シャンプーなどの多種類の物体である場合、画像内にそれらがあるかどうかを判断します。ある場合は、各物体の輪郭をマークし、対応するラベルを分類します。それがない場合、結果は表示されません。



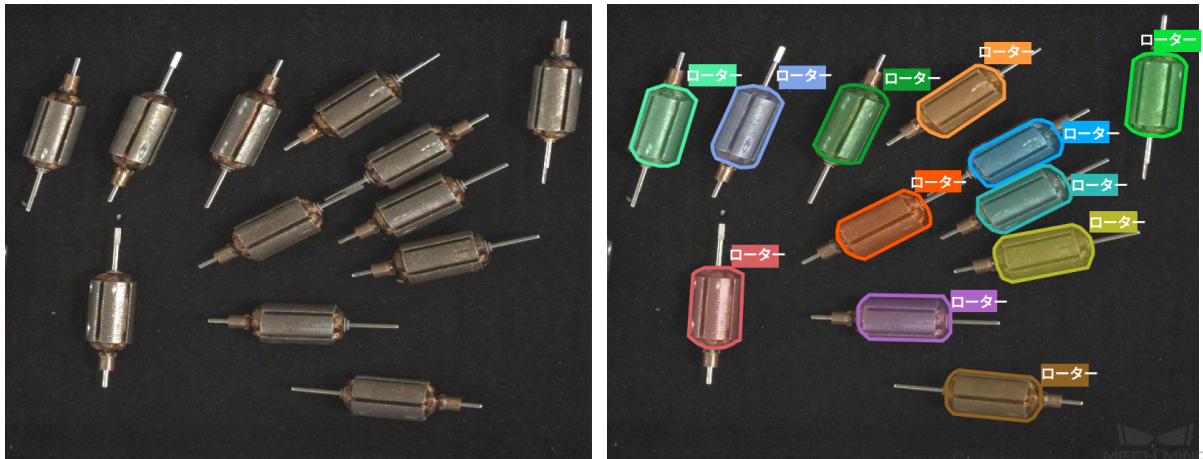
業界の代表的な応用シーン

- デパレタイジング・パレタイジング：紙箱、ターンオーバーボックス、麻袋などの物体を要件に従ってパレットから下ろし、パレットまたは破袋機、コンベヤーベルトなどに配置します。



麻袋の分割

- ワークのロードとアンロード：自動車、鉄鋼、機械などの業界で、複雑なワーク、構造部品、不規則な部品などの物体を運送、把持します。



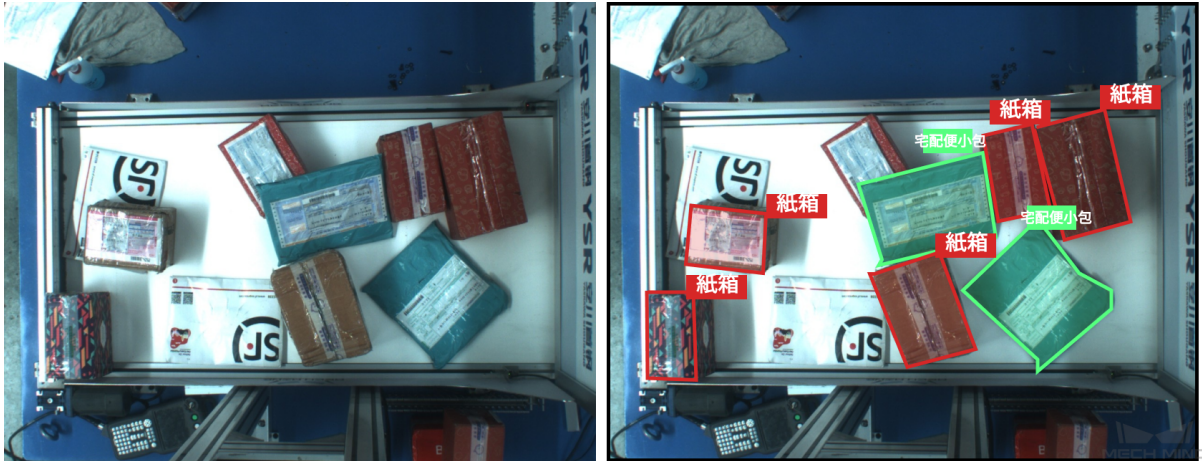
ワークの分割

- **商品仕分け**：種まき、摘み取りなどの EC 物流倉庫の商品仕分け方式に適しています。インフレーターバッグ、透明パッケージ、アルミニウム缶、不規則な商品（鍋、丼鉢などの食器）など、様々な商品に対応できます。



商品の分割

- **宅配便小包仕分け**：宅配便軟包、郵便封筒、宅配便の紙箱、フォーム封筒など一般的な小包、また様々な特殊形状の小包に対応できます。



紙箱と封筒の分割

応用プロセス

インスタンスセグメンテーションはディープラーニングに基づいています。ユーザーは、実際の応用シーンの物体画像をインスタンスセグメンテーションモデルに入力し、画像内の各対象物の輪郭とカテゴリをマークすることでモデルに分割方法を学習させます。ディープラーニングのインスタンスセグメンテーションの応用プロセスは以下のとおりです。

- **トレーニングに必要なデータを取得**：カメラで対象物の画像を十分に取得します。
- **データラベリング**：取得した画像ごとに、各物体の輪郭とカテゴリをマークします。
- **モデルトレーニング**：前の2つのステップで取得したラベル付きデータをインスタンスセグメンテーションモデルに入力してトレーニングを開始します。
- **モデルを使用して予測**：トレーニング済みモデルをプロジェクトに応用します。

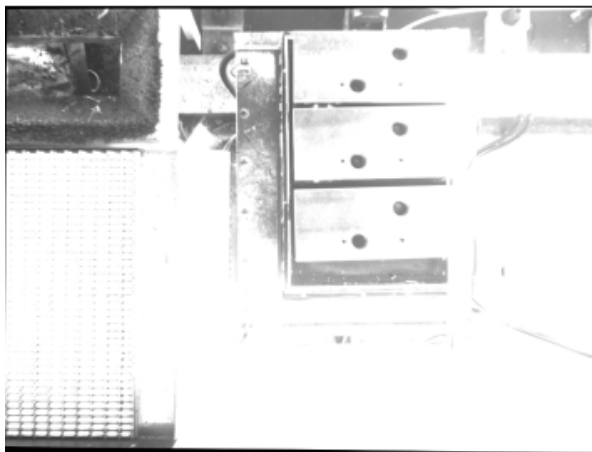
2.1.2 トレーニングに必要なデータを取得する

注意：ディープラーニングプロジェクトでは、最も重要なのはデータの取得です。モデルの最終的な効果は、トレーニング済みデータの品質に大きく依存します。高品質のデータセットを使用すれば、モデルトレーニングの品質と予測精度を向上させることができます。

データ取得環境を整備する

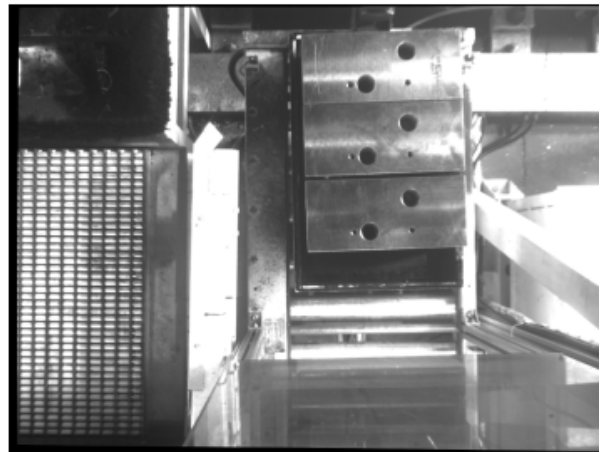
1. **露出過度、露出不足、色ずれ、ぼやけ、遮蔽**などを避けてください。これらが発生したら、ディープラーニング・モデルに必要な特性がなくなり、モデルの効果に差し支えます。

間違った例：露出過度



最適化ソリューション：遮光などの方法で回避できます

正しい例：適正露出



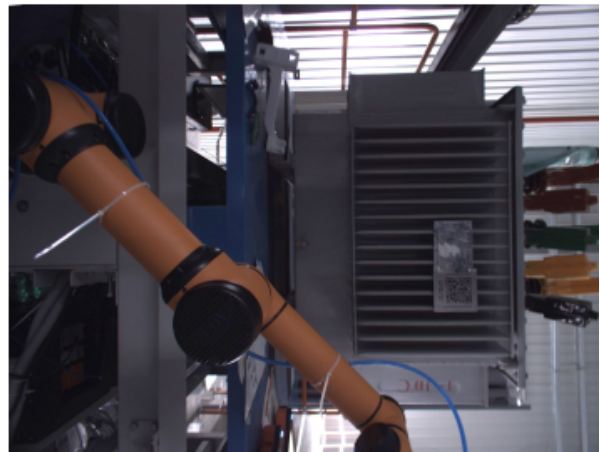
MECH MIND

間違った例：露出不足



最適化ソリューション：補光などの方法で回避できます

正しい例：適正露出



MECH MIND

間違った例：色の歪み



最適化ソリューション：カメラのホワイトバランスを調整することで回避できます

正しい例：正常



間違った例：ぼやけ

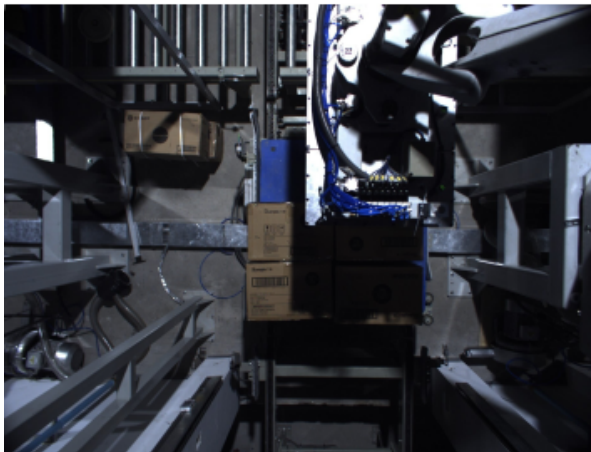


最適化ソリューション：物体が移動中に撮影しないでください

正しい例：正常



間違った例：ロボットアームがカメラの視野を遮る



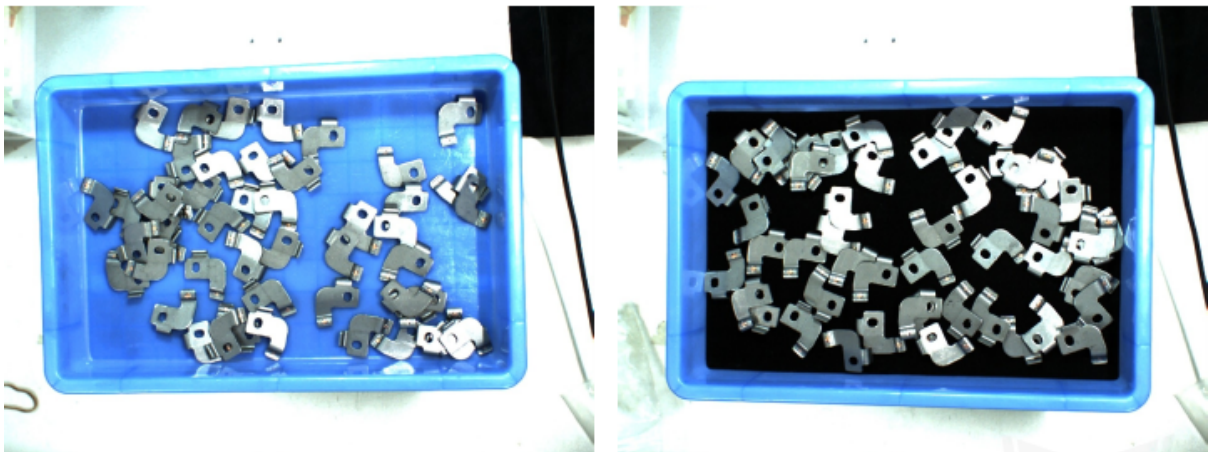
間違った例：人間がカメラの視野を遮る



最適化ソリューション：ロボットアームと人間が撮影範囲外にあることを確認してください

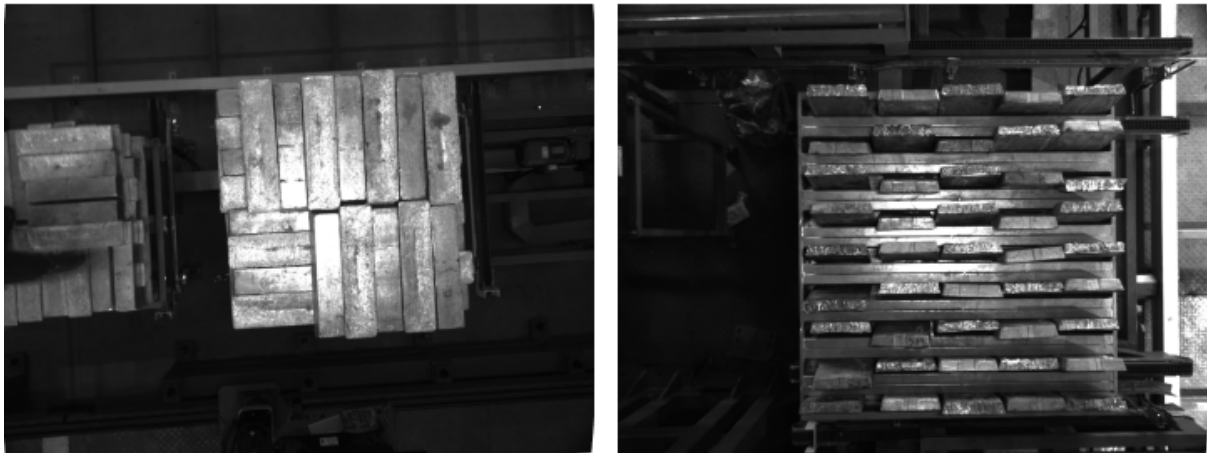
- 取得するデータの背景、視角、高さが実際の応用と一致するようにしてください。いかなるズレがあっても、実際応用のディープラーニングの効果が低下し、改めてデータを取得しなければならないこともあります。したがって、必ず事前に実際応用の状況をご確認ください。

間違った例：トレーニングデータの背景（左）が実際の物体背景（右）と不一致



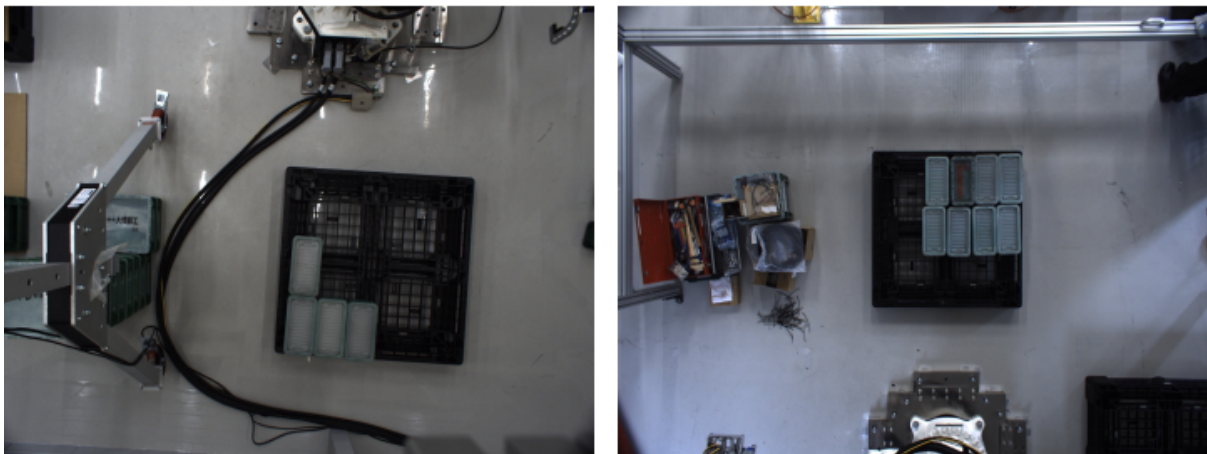
最適化ソリューション：トレーニングデータの背景を実際の物体背景と一致させてください

間違った例：トレーニングデータの取得視野（左）が実際の物体取得視野（右）と不一致



最適化ソリューション：トレーニングデータの取得視野を実際の物体取得視野と一致させてください

間違った例：トレーニングデータを取得したカメラの高さ（左）が実際応用中のカメラの高さ（右）と不一致



最適化ソリューション：トレーニングデータを取得したカメラの高さを実際応用中のカメラの高さと一致させてください

取得する画像枚数

- 対象物が単一種類である場合、約 50 枚画像を取得します。
- 対象物が多種類ある場合、各種類に対して約 30 枚を取得します。取得合計数は、「30 * 種類」です。
- 上記は一般的な数で、業界の代表的な応用ではより詳細な基準があります。詳しくは、[代表的なプロジェクトのデータ取得の実例](#) をご参照ください。

注意： トレーニングデータが少なすぎると、モデルに必要なサンプルは足りません。それに基づいてフィッティングされたデータを使用すれば、ディープラーニングモデルを効果的にトレーニングすることができなく、テストのエラー発生率も高くなります。一方、データが多すぎると、トレーニングの速度が遅くなります。以上のことから、必ず適切に取得してください。

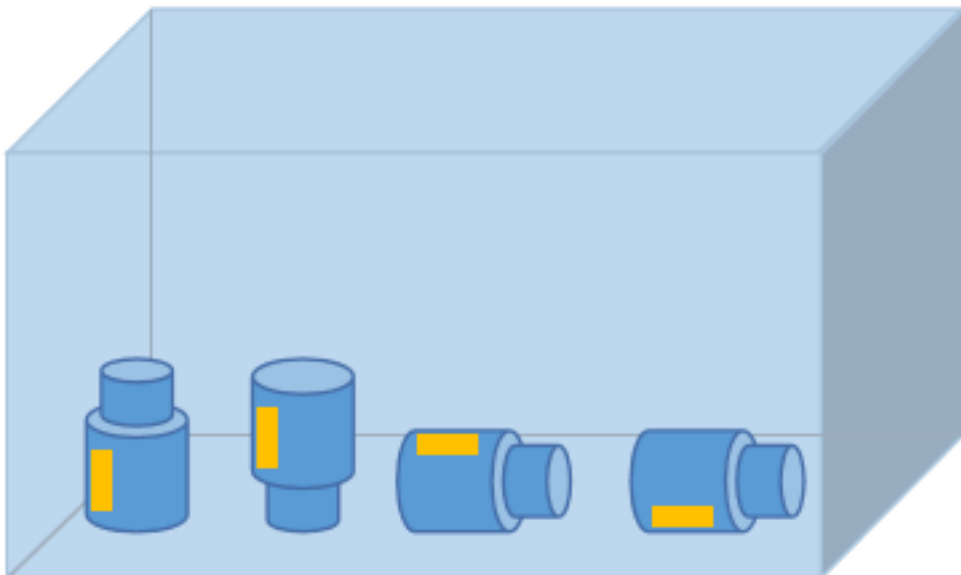
取得時の物体姿勢

物体の様々な配置状況に対して、取得要件に応じて合理的に取得する必要があります。たとえば、生産現場では、供給するワークは水平方向と垂直方向があるとします。水平方向のデータだけを取得してトレーニングすれば、垂直方向のワークに対する認識効果は保証できません。また、単一ワークのデータだけを取得すれば、積み重ねられて遮蔽されたワークに対する認識効果も保証できません。なので、データを取得するときは、**実際の生産現場の様々なシーンを含める** 必要があります。詳しくは：

- 取得したデータに、実際の応用シーンにありうる **物体の向き** が含まれていることを確認してください。
- 取得したデータに、実際の応用シーンにありうる **物体の位置** が含まれていることを確認してください。
- 取得したデータに、実際の応用シーンにありうる **物体間の関係** が含まれていることを確認してください。

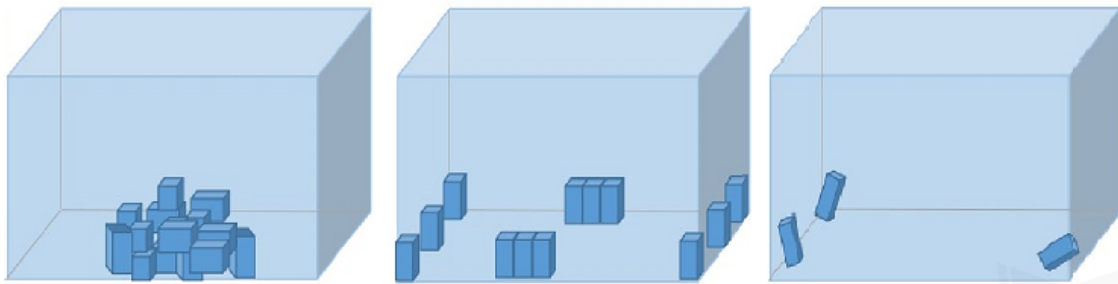
注意： ある状況の取得が脱落したら、その状況のディープラーニングは実行できず、認識することもできません。エラー発生率を下げるために状況に応じてデータサンプルを追加してください。

1. 物体の向き



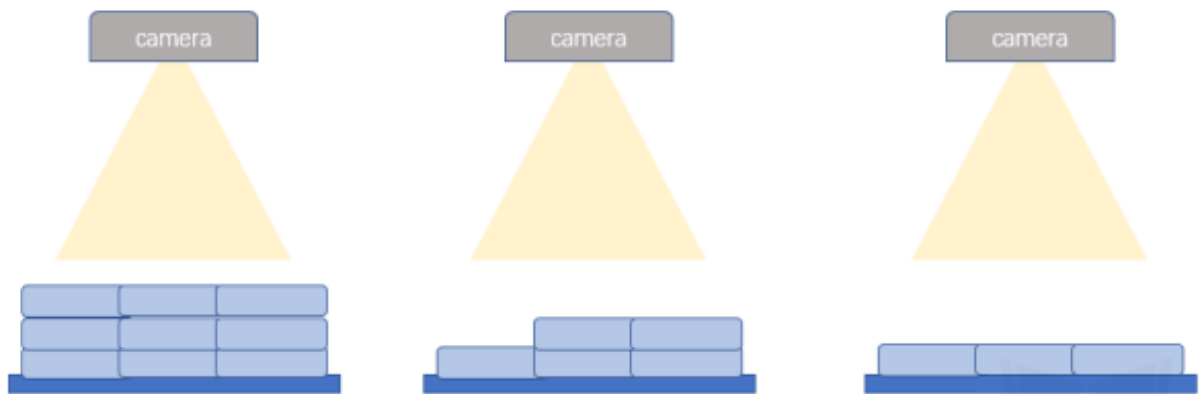
異なる面が上向き

2. 物体の位置



MECH MIND

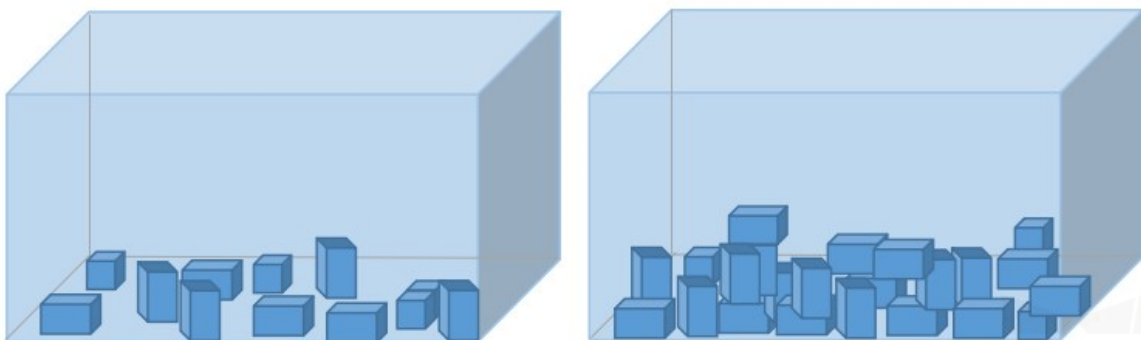
ボックスの中央、端、コーナーにある



MECH MIND

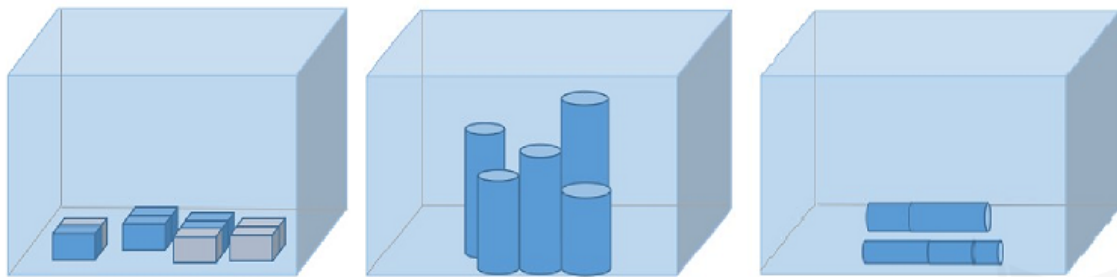
パレットの異なる段にある

3. 物体間の関係



MECH MIND

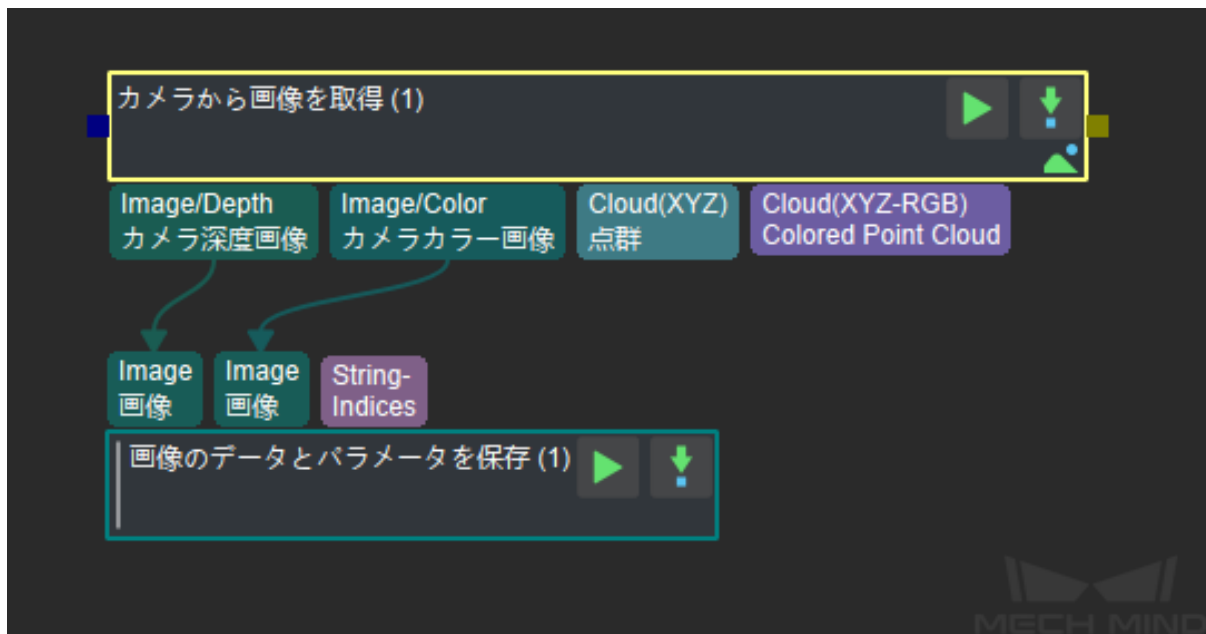
接触しない、積み重ね



密接

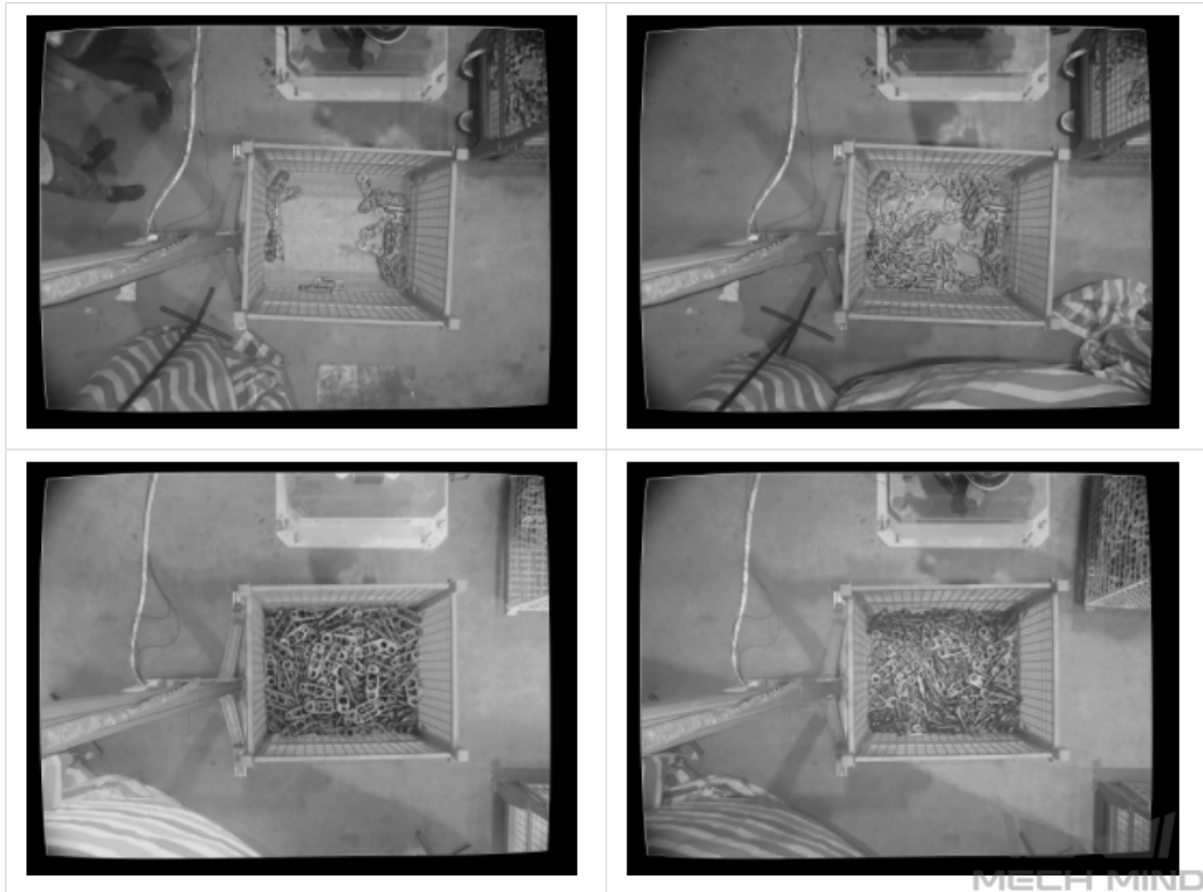
Mech-Vision を使用してデータを取得する

上記の取得環境、取得数、取得時の物体配置状況を確認した後、Mech-Vision で下図に示すプロシージャを構築してデータを取得できます。ステップの使用については、capture_images_from_camera をご参照ください。

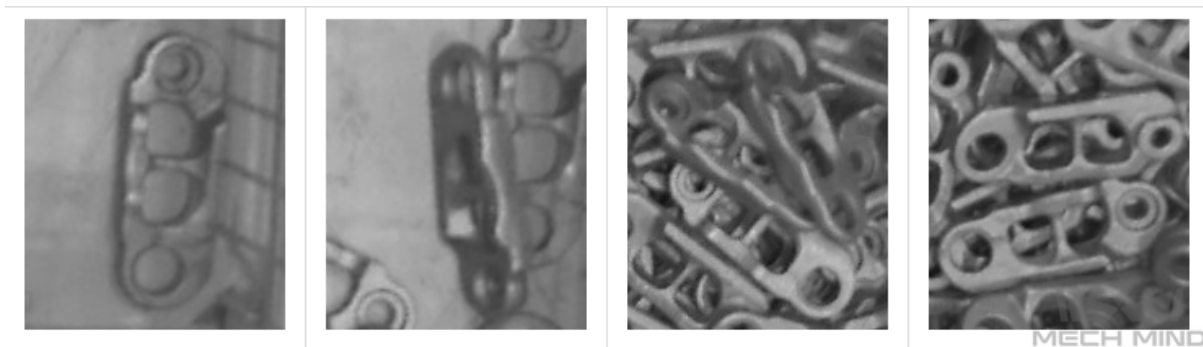


プロジェクトにおけるデータ取得の実例

1. ある金属部品プロジェクトでは、対象物は単一種類なので、画像を 50 枚取得します。物体の向きについては、平らに放置しているものもあれば、横に立っているものもありますので、取得するときにこれを考慮に入れます。物体の位置については、ボックスの中央、端、コーナーにある状況や高さも考慮する必要があります。物体間の関係については、積み重ねる状況に加えて、一部が平行に並んでいる状況を考慮する必要があります。実際に取得された画像は以下のとおりです。



少量のバラ積み物体（左上） / 集中のバラ積み物体（右上） / 積み重ねる物体（左下） / 集中のバラ積み物体（右下）



平らに放置 / 横に立っている / 積み込める / 並んで放置

- ある日用品プロジェクトでは、7種類の物体があって、これらを分類します。物体特性のデータを完全に取得するには、「単一種類の物体が異なる向きに配置される」と「複数種類の物体が混合配置される」の状況を考慮する必要があります。単一種類物体の画像を取得する際、取得枚数は「5 * 種類数」です。複数種類の物体の画像を取得する際、取得枚数は「20 * 種類数」です。物体の向きについては、平らに配置され、横に立ち、傾いているという姿勢があり、取得するときはいちいち考慮する必要があります。物体の位置については、ボックスの中央、端、コーナーにある状況も全部考慮します。物体間の関係については、積み重ねのほかに、平行に並んでいる、密接するなども考慮する必要があります。実際に取得された画像は以下の通りです。

単独放置：



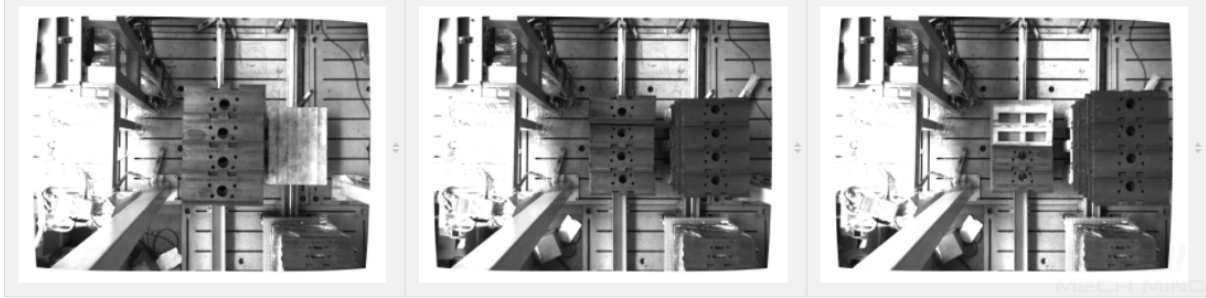
ボックスのコーナーにある（左上）/密接（右上）/密に積み込まれ（左下）/少量のバラ積み物体（右下）

混合放置：



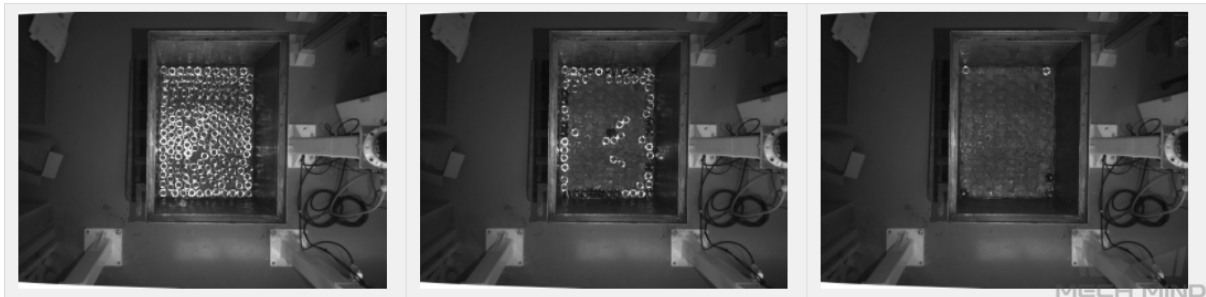
密に積み込まれ / ボックスのコーナーにある / バラ積み

- あるトラックシューズのプロジェクトでは、複数型番の部品があるため、画像の取得枚数は「30*型番の数」です。物体の向きについては、正面が上向き状況のみを考慮します。物体の位置については、配置方式は同じく、高・中・低層の画像データをそれぞれ取得する必要があります。物体間の関係については、規則的に配置されているため、密接した部品に注意を払う必要があります。実際に取得された画像は以下の通りです。



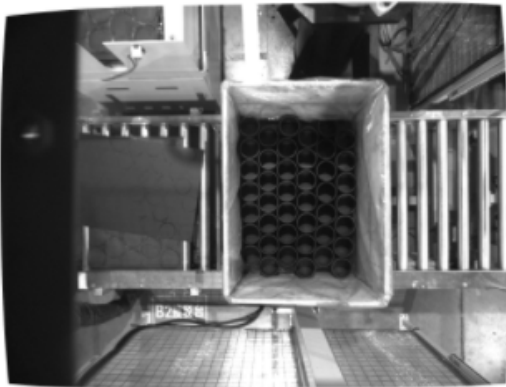
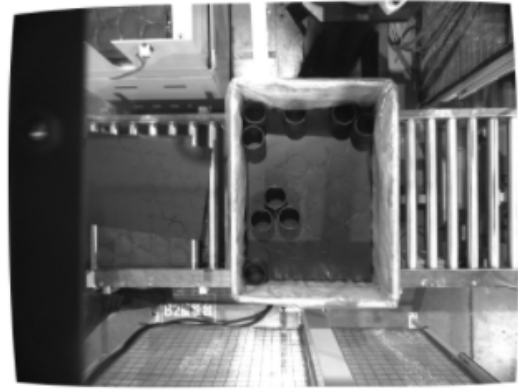
異なる高度の視野（高、中、低）

- ある金属部品プロジェクトでは、部品が一層に並んでいるので、50枚の画像を取得します。物体の向きについては、正面が上向きの画像だけを取得します。物体の位置については、ボックスの中央、端、コーナーに位置する画像を取得します。物体間の関係については、密接した状況を考慮する必要があります。実際に取得された画像は以下の通りです



段ごと / ボックスの端にある / ボックスのコーナーにある

- ある金属部品プロジェクトでは、物体が規則的に多層に積み重ねられているため、30枚の画像を取得します。物体の向きについては、正面が上向きの状況だけを考慮する必要があります。物体の位置については、ボックスの中央、端、コーナーに位置する画像と高・中・低層の画像を取得します。物体間の関係については、密接した状況を考慮する必要があります。実際に取得された画像は以下の通りです。



高層にある（左上）/少量で高層にある（右上）/低層にある段ごと（左下）/ボックスの端にある（右下）

2.1.3 データラベリング

ラベルを作成する

プロジェクトでは分類が必要かどうかを確認します。分類が必要な場合、カテゴリに対応する数のラベルを作成します。分類が不要な場合は、ラベルを一つ作成します。

ラベルを作成

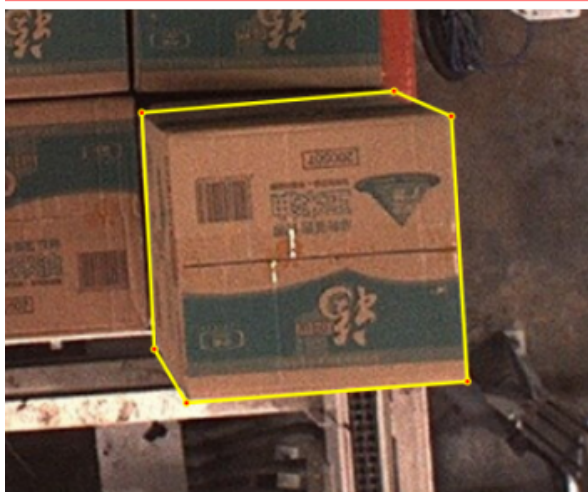


注意: ラベル名は認識しやすいものにしてください。a、b、tmp など意味のない名前は使用しないでください。ラベル名には英語の文字と数字以外の文字を使用しないでください。あいまいさを避けるために、日本語ローマ字 (hako など) ではなく、英語の単語 (box など) を使用してください。

ラベリング方法を確認する

1. **表面輪郭のみをマーク:** 紙箱、葉箱、長方形ワークなど、平らに配置された規則的形狀の物体に適用されます。上表面から把持するため、側面をマークする必要はありません。側面もマークすれば、トレーニングの効果に差し支えます。

間違った例：外部輪郭をマーク

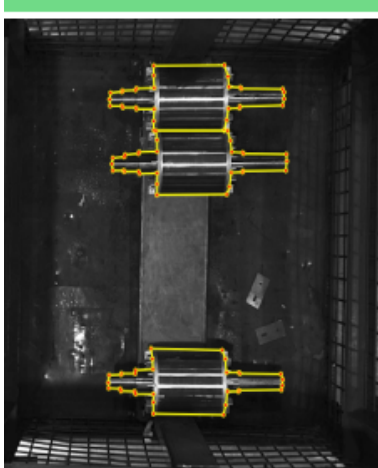


正しい例：表面輪郭のみをマーク



2. **完全な輪郭をマーク:** 一般的方法で、麻袋、各種ワークなどに適用されます。

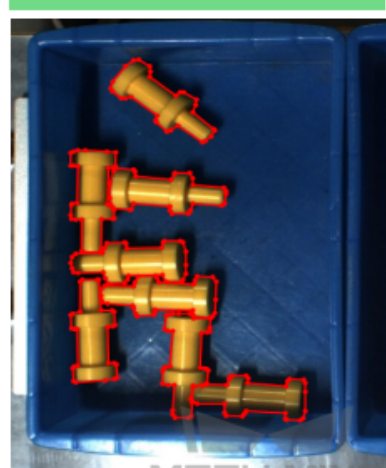
正しい例：完全な輪郭をマーク



正しい例：完全な輪郭をマーク



正しい例：完全な輪郭をマーク



3. 特別な状況：治具や把持方法と併用する必要がある場合に適用されます。

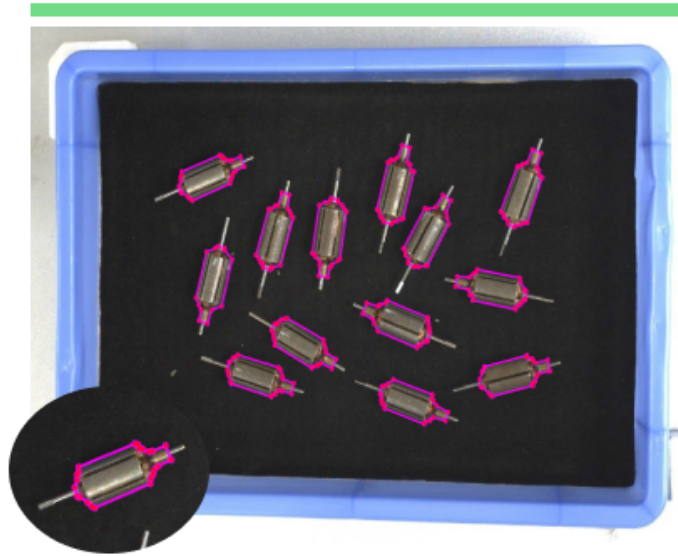
実例 1：吸盤がボトルの口を密に吸着するように（高精度が必要）、ボトル本体ではなく、ボトルの口だけをマークします。

正しい例：ボトルの口をマーク



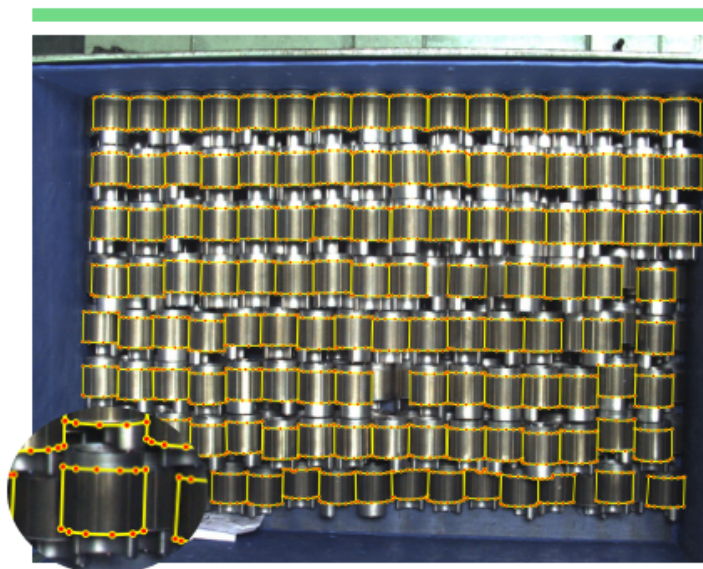
実例 2：ローターの把持には方向を区別する必要があるため、方向が明確に識別できる中央部分のみをマークし、両端の細いロッドをマークしません。

正しい例：ローターの中央部分をマーク



実例 3：金属部品の中央を吸着するように、部品の中央のみをマークし、両端をマークしません。

正しい例：中央部分をマーク



Mech-DLK を使用してデータをラベリングする

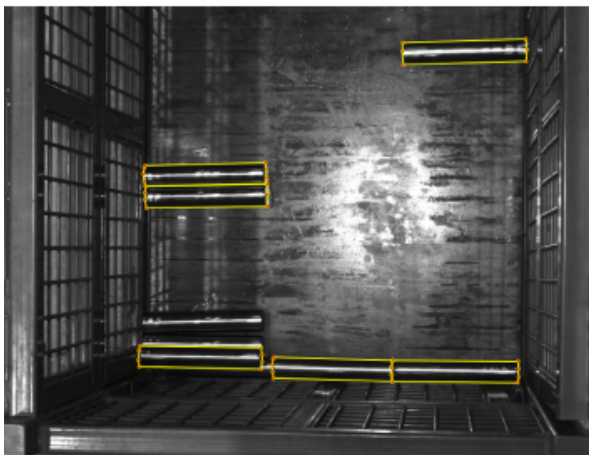
アルゴリズムによってラベリング方法が異なります。詳しくは、Mech-DLK データラベリング方法 をご参照ください。

注意: ラベリングの品質を確認：ラベリングした後、必ずデータセットのラベリング状況を確認し、ラベルが各サンプルに対応しているかどうかを確認してください。ラベル分類が間違い、データセットで反例を使用すれば、ラーニングのプロセスではモデル認識効果に直接影響します。

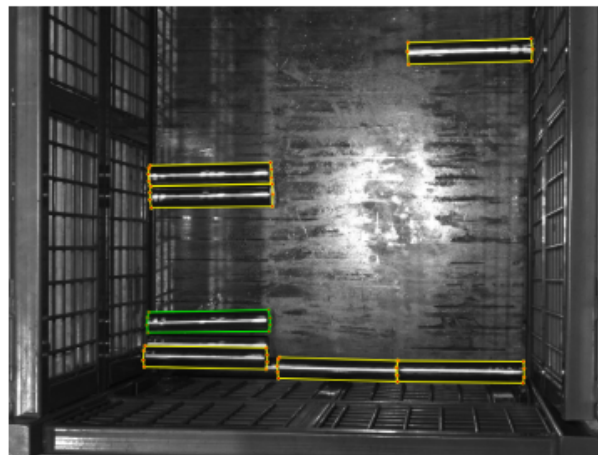
ラベリングの品質は、完全性、正確性、一貫性、精確性が含まれます。

1. **完全性**：全ての対象物をラベリングします。

間違った例：ラベル付けの欠落



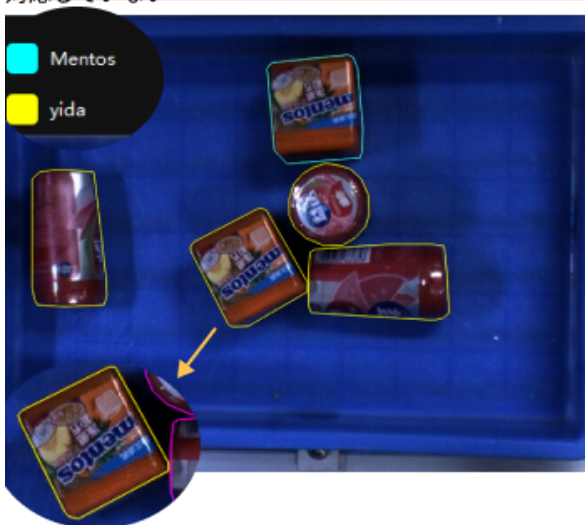
正しい例：正しいラベル付け



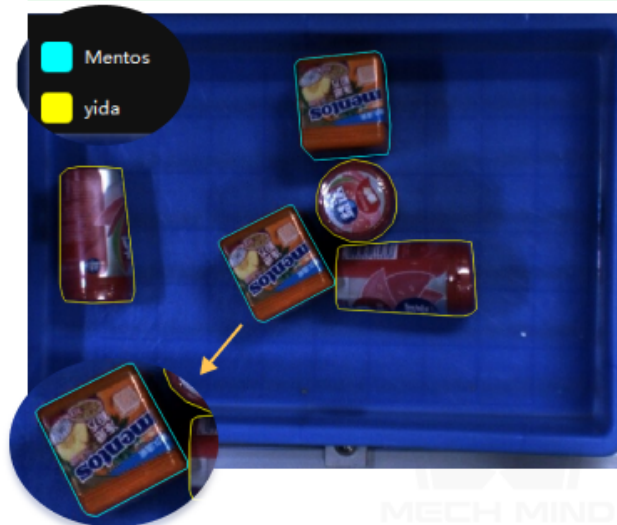
注意：すべての完全な棒鋼にラベル付けが必要な場合は、欠落があってはなりません。

2. **正確性**：物体とラベルとは正確に対応するようにします。

間違った例：Mentosはyidaとしてマークされ、ラベルは対応していない

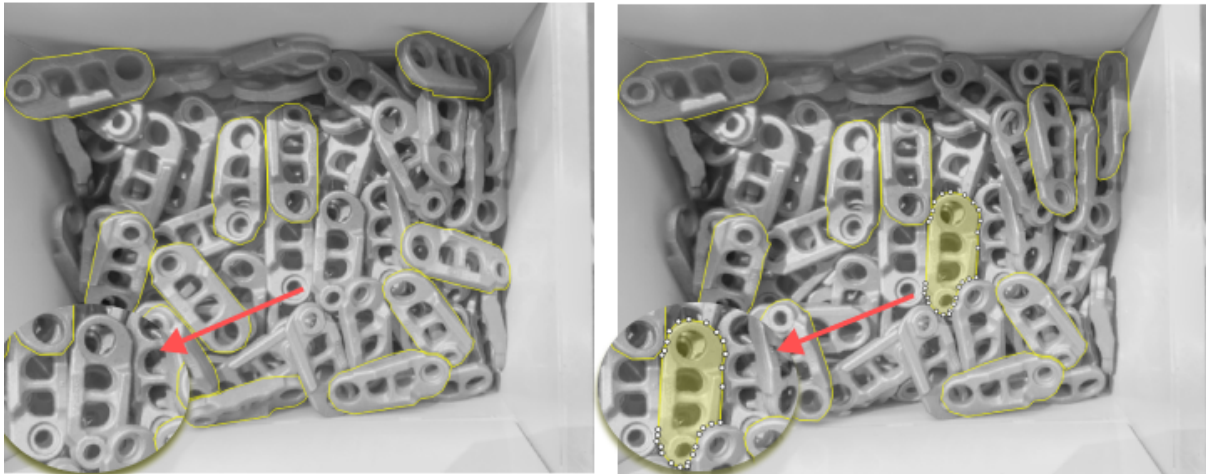


正しい例：正しいラベル付け



3. **一貫性**：全てのデータに対して同じ基準に従ってラベリングします。たとえば、全体 85 %以上が露出した物体のみをラベリングするという基準であれば、基準を満たす全ての物体をラベリングする必要があります。

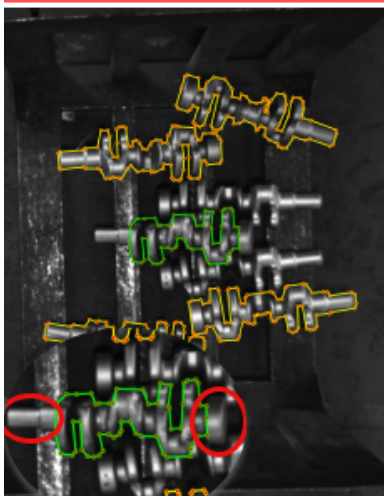
間違った例：隣接する二つの画像に同じ姿勢の金属部品のラベルが不一致



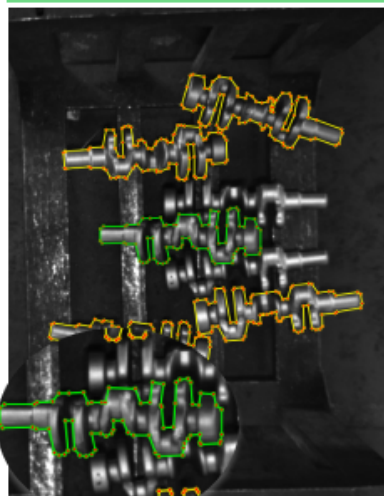
注意：ラベルが不一致な場合は、異なるところのラベルが要件を満たしているかどうかを確認してください。要件を満たしている場合は、欠落しているラベルを補足してください。そうでない場合は、多すぎるラベルを削除してください。

4. **精確性**：マークされた輪郭は対象物のエッジとぴったりと一致するようにします。物体の一部が脱落したり、余計な部分が入ったりしないようにラベリングを行ってください。

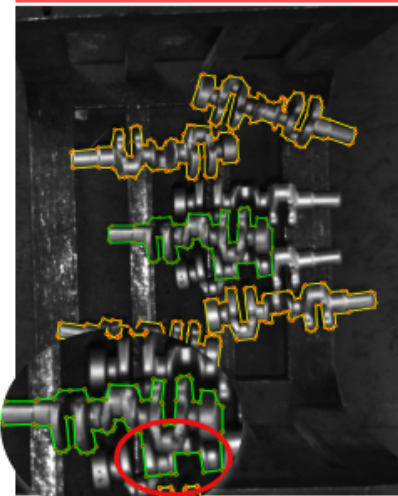
間違った例：ラベル付けが欠落



正しい例：正しいラベル付け



間違った例：ラベル付けが多すぎる



注意：すべてのクランクシャフト全体にラベルを付ける必要がある場合は、ある部分のラベルが欠落または多すぎないでください。

2.1.4 モデルトレーニング

Mech-DLK を使用してモデルをトレーニングする

詳しくは、[クイックスタート](#) をご参照ください。

調整可能なトレーニングパラメータ

通常、デフォルトのパラメータを使用してトレーニングすれば要件を満たせます。特別なニーズがある場合に以下のパラメータを調整します。

パラメータ
トレーニング
検証

データ拡張パラメータ ▼

輝度 -60 % ~ 60 %

コントラスト -60 % ~ 60 %

並進 -15 % ~ 15 %

回転 -135 ° ~ 135 °

スケーリング -50 % ~ 50 %

画像反転

トレーニングパラメータ ▼

エポックの合計数 < 200 >

学習率 < 0.0050 >

応用
リセット

トレーニング
検証

輝度範囲

現場で照明の変化が大きく、補光や遮光でも改善できない場合は、取得したトレーニングデータの差異も大きいので、トレーニングのために輝度範囲を適切に大きくします。照明の変化が大きい場合は、デフォルトのパラメータを使用します。

コントラスト範囲

物体と背景の違いが明らかではない場合、モデルに物体の特性を学習させるために、範囲を適切に大きくすることができます。通常、輝度範囲と組み合わせて調整します。普通、これを調整する必要はありません。

並進範囲

物体の背景（ボックス、パレットなど）の移動範囲が大きい場合は、並進範囲を大きくする必要があります。大きくない場合は、デフォルトのパラメータを使用します。

回転範囲

物体が固定位置にあり、向きを判別する必要がある場合、回転によるモデル学習方向特性への影響を回避するように、範囲を「0-0」に調整してください。それ以外の場合は、デフォルトのパラメータを使用します。

スケール範囲

物体の位置高度の違いが大きいまたは物体が同じ高さであり、ボリュームの違いが大きい場合、スケール範囲を大きくする必要があります。それ以外の場合は、デフォルトのパラメータを使用します。

画像反転

物体が固定位置にあり、向きを判別する必要がある場合、反転によるモデル学習方向特性への影響を回避するように、「画像反転」オプションをオフにしてください。通常、回転範囲と組み合わせて調整します。それ以外の場合は、デフォルトのパラメータを使用します。

チャンネルシャッフル

色によって分類し、形状の違いが大きい場合、混色の変化によるモデルの物体の色特性の学習に影響しないように、このオプションをオフにしてください。それ以外の場合は、デフォルトのパラメータを使用します。

エポックの総数

分類する必要がなく、対象物の特性が簡単な場合、「600epoch」以内に最適モデルを選択するよう設定できます。物体の姿勢が複雑で、種類が多い場合、または他の特性が複雑な場合、「1000epoch」以内に最適モデルを選択するよう設定できます。

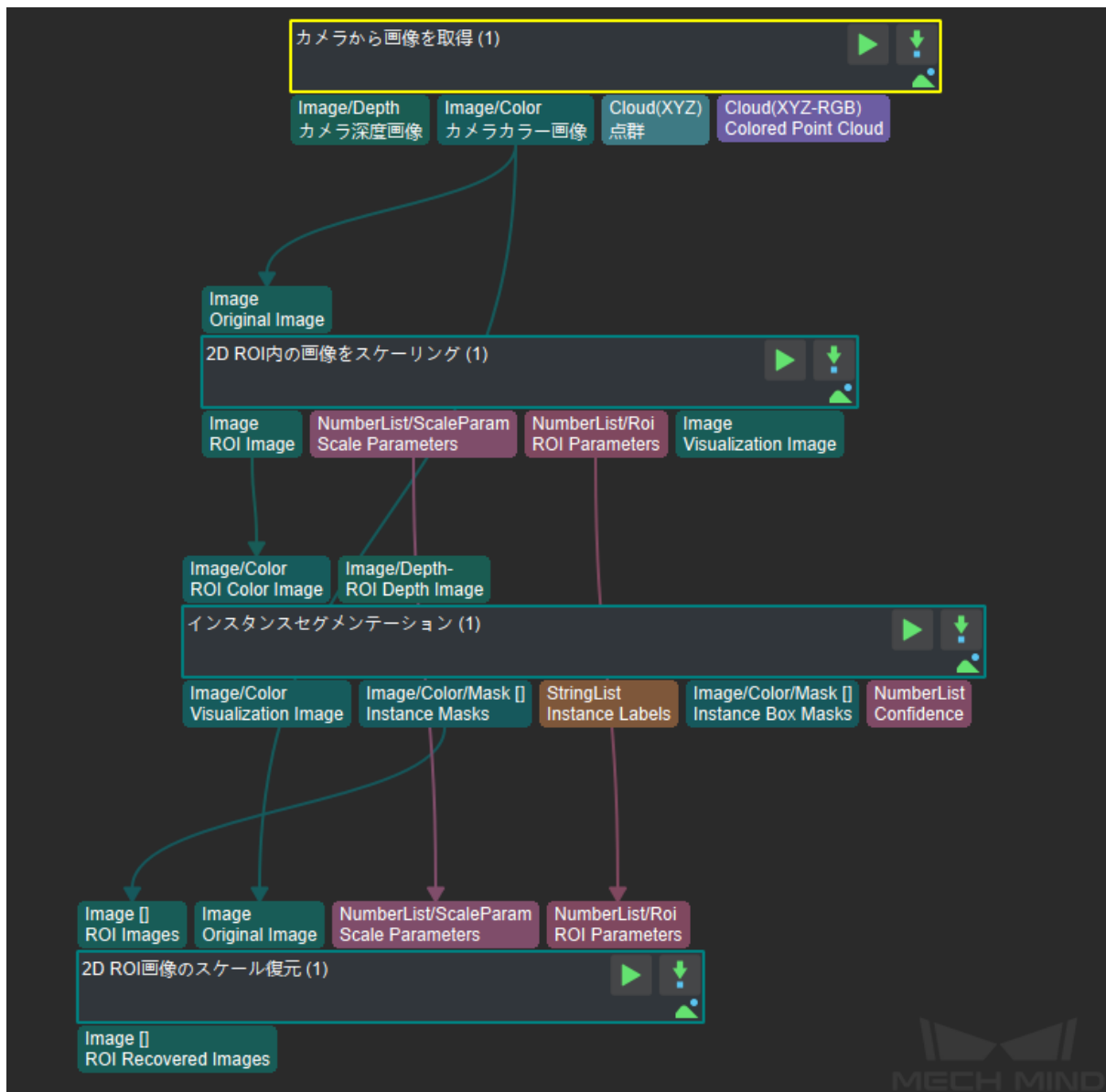
学習率

通常、調整する必要はありません。正確率が非常に低く (<0.8) になった場合、現在の学習率が高いからかもしれないのでそれを 10 分の 1 に調整してください。

注意: 実際に利用できない範囲に調整すれば、モデルの効果に影響します。たとえば、照明が安定している場合、余計に輝度範囲を調整するより、デフォルトのパラメータを使用したほうが良い効果が得られます。

2.1.5 モデルを使用して予測する

Mech-Vision でインスタンスセグメンテーションモデルを使用する



ステップ1: 「カメラから画像を取得」を実行します。

ステップ2: 「2D ROI内の画像をスケーリング」し、実際のデータをトレーニングデータのROIと一致させます。ROIの手動・自動スケーリング設定に注意する必要があります。通常、デフォルトの自動スケーリングを使用します。

属性	値
ステップ名	2D ROI内の画像をスケーリング_1
▶ 実行フラグ	
画像入力ポート数	1
▶ カラー画像のROI設定	ROIByFile ★
カラー画像のROIエリアの更新方法	Origin ▼
▼ カラー画像のROIスケーリング設定	
自動スケーリング	False
カスタムスケールファクタ	0.6400
▼ 出力画像の理想的な長さとの設定	
入力カラー画像のROIと同じ長さとの幅 (スケーリングなし)	False
水平方向の長さ	1024 pixel
垂直方向の幅	1024 pixel
▼ 画像を目標サイズまでパディング	
パディング	✓ True
R (0 ~ 255)	0
G (0 ~ 255)	0
B (0 ~ 255)	0

ステップ3:「インスタンスセグメンテーション」によりディープラーニングのセグメンテーション結果が出力されます。セグメンテーションの結果が要件を満たすかどうか、マスクが完全であるかどうか、認識に誤りや漏れが発生したかどうかを判断できます (モデルファイル.pth とコンフィグファイル.py を正しく入力してください)。以下のパラメータに注意してください。

属性	値
ステップ名	インスタンスセグメンテーション_1
▶ 実行フラグ	
▼ サーバー & モデルの設定	
サーバーIP	127.0.0.1
サーバーポート (1 ~ 65535)	50052
モデルファイル	D:/...
コンフィグファイル	D:/...
▼ プリロード設定	
プロジェクトを開くとモデルを自動プリロード	False
最大検出対象物の数 (0~2000)	500
信頼度しきい値 (0 ~ 1.0)	0.7000
▼ フォント設定	
カスタムフォントサイズを使用	✓ True
フォントサイズ (0 ~ 10)	2.0
▼ 可視化設定	
画像で検出する対象物を描画	✓ True
異なる対象物の可視化方法	Classes ▼

プリロード設定-最大検出対象物の数

モデルが一回で数十または数百の物体を認識する場合は、最大検出対象物の数を調整してください。

対象物の数が少なくかつプロジェクトのタクトが指定された場合は、このパラメータを小さくすれば、タクト向上を実現できます。

プリロード設定-信頼度しきい値

普通、信頼度しきい値は、デフォルトの 0.7 にすればいいです。認識された物体のマスクの信頼度がデフォルトより低ければ、この物体を把持できないと判断し、信頼度の値が赤字で表示されます。プロジェクトの精度に対する要求に従ってこのパラメータを調整できます。例えば、信頼度の値が低くて物体のマスクが完全で把持できると判断した場合に、信頼度のパラメータを適切に低く調整することで把持可能な物体数を増やします。

フォント設定-カスタムフォントサイズを使用

通常はデフォルトのフォントサイズを使用すればいいです。認識対象物が小さければ、効果を確認しやすいようにフォントサイズを適切に小さく調整できます。

可視化設定-画像で検出する対象物を描画

「True」をチェックにすれば認識対象物のマスクを認識でき、それでモデルの認識効果を確認できます。実際に実行する時はオフにしてタクト向上が可能です。

可視化設定-異なる対象物を可視化する方式

「Classes」を選択すると、色が異なるマスクで異なるラベル種類を表示します。「Instances」を選択すると、色が異なるマスクで異なる物体を表示します。「Threshold」を選択すると、色が異なるマスクで信頼度しきい値より高い物体と低い物体を表示します。

ちなみに: あるプロジェクトで2つのインスタンスセグメンテーションモデルが必要な場合は、2つの「インスタンスセグメンテーション」ステップを使用することになり、対応するモデルファイルとコンフィグファイルを設定します。さらに、**サーバー IP アドレス** をそれぞれ「127.0.0.1:50052」と「127.0.0.1:50053」に設定する必要があります。そうしないと、ポートの衝突が発生します。

ステップ 4: 「2D ROI 画像のスケール復元」を実行します (画像を元のサイズに戻します)。

2.2 画像分類

2.2.1 全体的な紹介

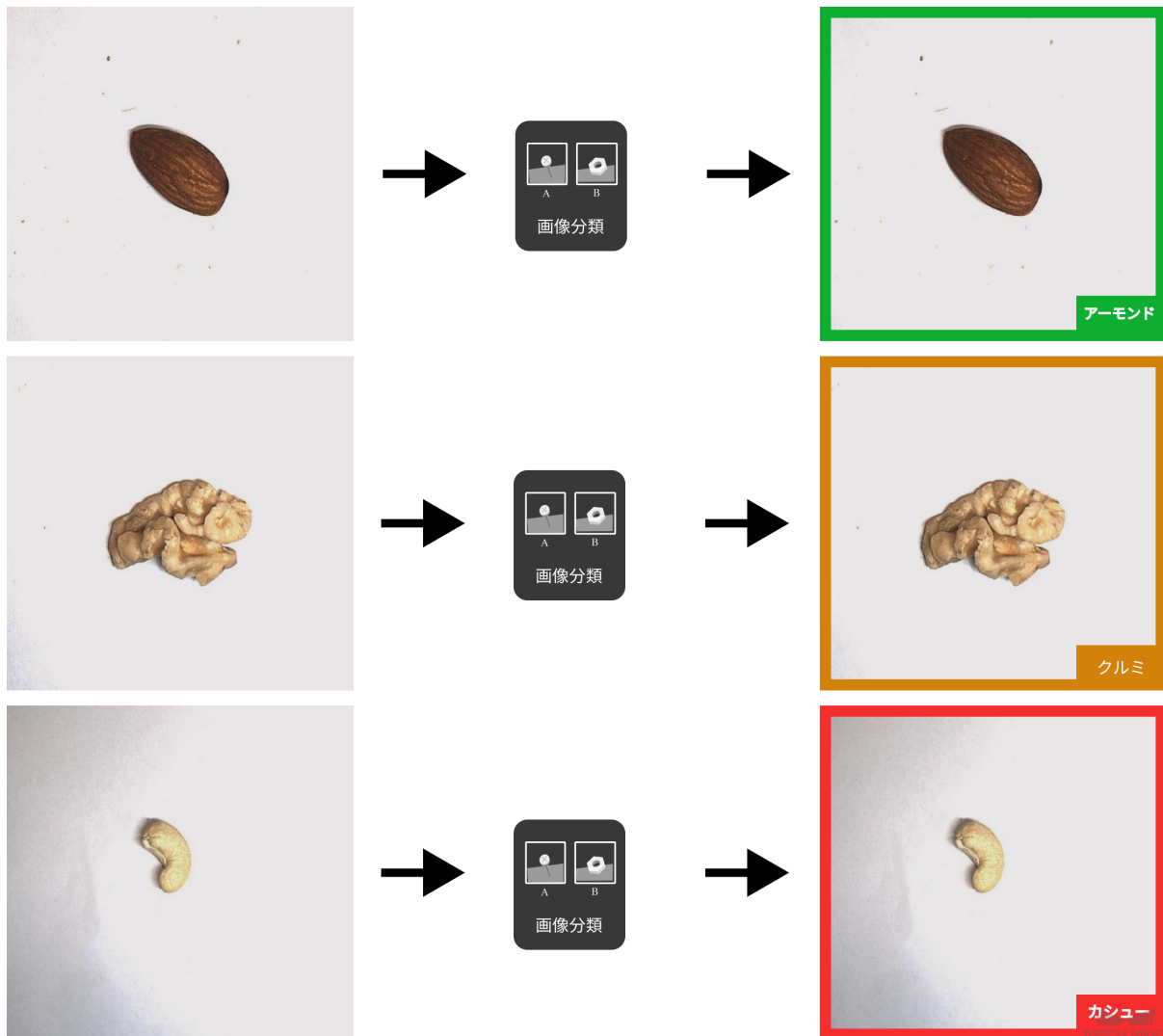
画像分類の機能

画像分類では、「これは何?」を回答します。

即ち、画像にある物体のカテゴリ、ワークのサイズと型番を認識し、物体の正面か背面か、正確な位置にあるかないかを判断します。

具体例:

対象物はアーモンド、クルミ、カシューです。画像を入力し、その中のものはどの種類の対象物であるかを判断し、対応するカテゴリラベルが出力されます。

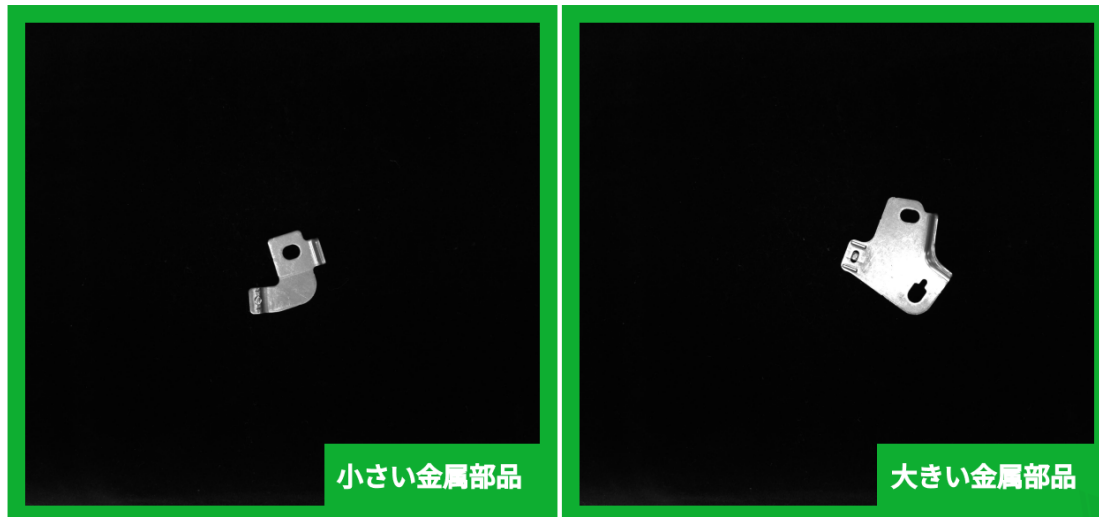


注意: 画像分類によって出力されたラベルは、画像全体を対象とします。画像に複数のカテゴリの物体があり、それらを分類する場合は、画像を分割する必要があります（より小さな画像に分割し、分割された画像に1つだけの物体があるようにします）。また、状況に応じて **インスタンスセグメンテーション** もしくは「対象物検出」を使用することもできます。

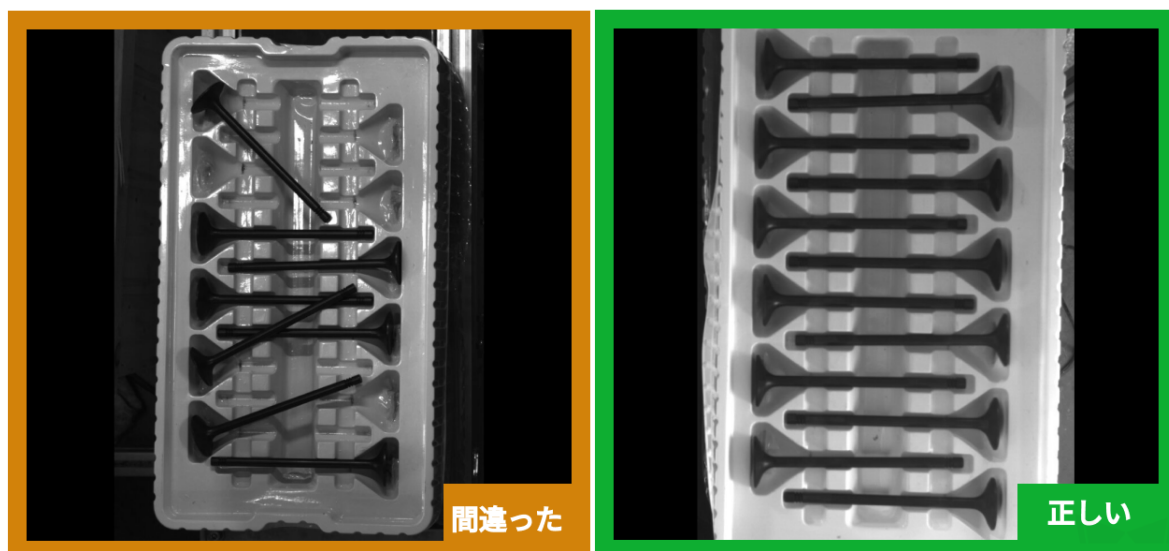
業界の代表的な応用シーン

画像分類モデルを使用すれば、プロジェクトで異なるカテゴリの画像を区別することができます。以下では、典型的な応用シーンを紹介します。

- ワークのロードとアンロードプロジェクトで、ワークの種類、向き、正面/背面を認識します。



- 組立/ピッキングのプロジェクトでは、物体が正確に配置されたかどうかなどを判断します。



応用プロセス

ディープラーニングに基づく画像分類では、ユーザーは実際応用の物体画像を十分に入力し、認識される物体が属するラベルカテゴリをマークする限り、画像分類モデルはその区別を学習できます。ディープラーニングの画像分類の応用プロセスは以下の通りです。

- **トレーニングに必要なデータを取得**：カメラで各カテゴリの物体の画像を十分に取得します。
- **データラベリング**：取得した各画像に対応するカテゴリラベルを設定します。
- **モデルトレーニング**：前の2つのステップで取得されたラベル付きデータを画像分類モデルに入力して学習を実行します。
- **モデルを使用して予測**：トレーニング済みモデルをプロジェクトに応用します。

2.2.2 トレーニングに必要なデータを取得する

データ取得環境を整備する

詳しくは、[インスタンスセグメンテーション：データ取得環境を整備](#) をご参照ください。

注意：画像分類では **照明** が大事なので、安定した照明を確保する必要があります。照明が変動する現場では、状況に応じて個別に取得してください。

取得する画像枚数

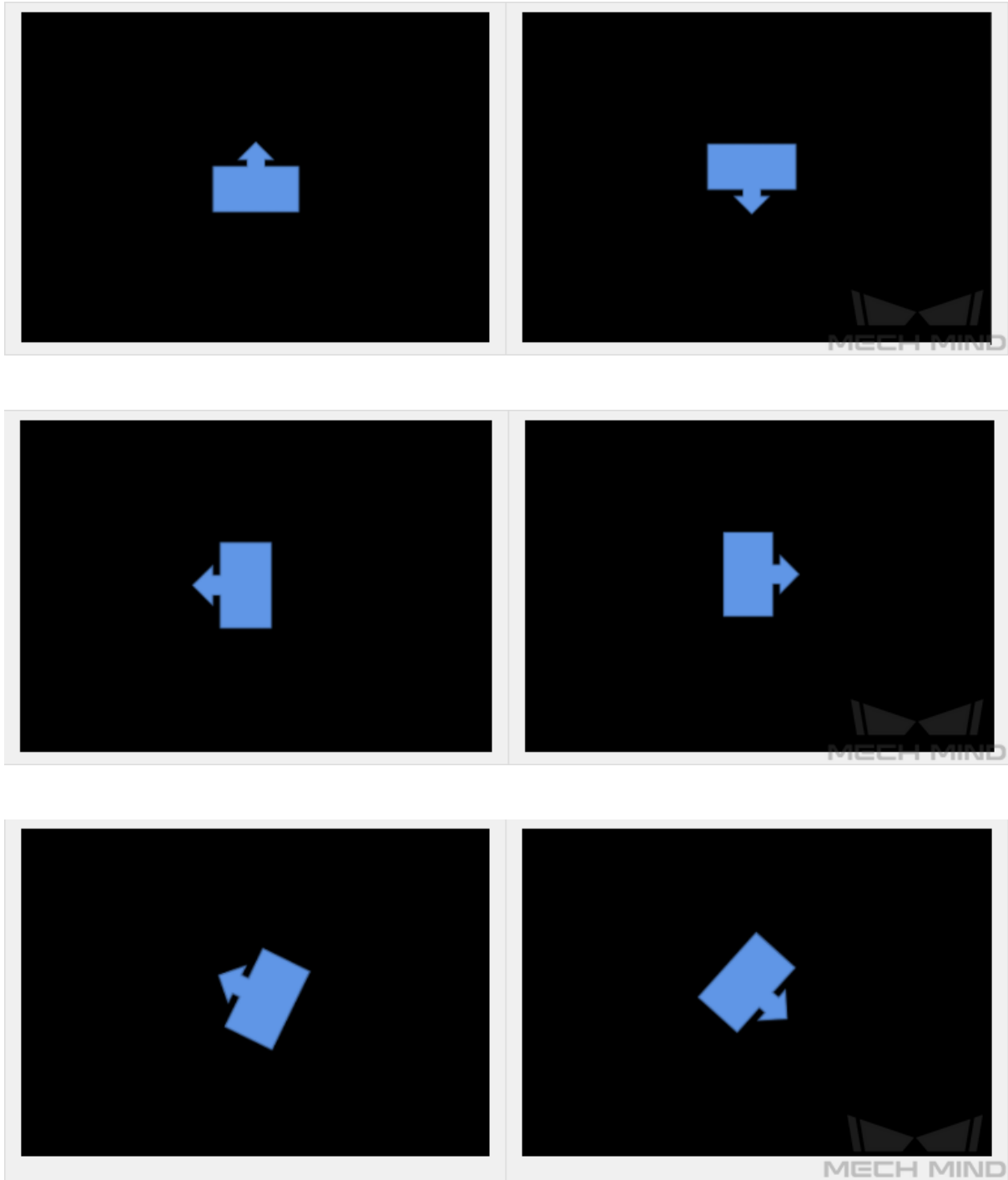
- 各カテゴリの物体に対して約 20 枚の画像を取得します。

取得時の物体姿勢

取得の要件に応じて様々な放置姿勢の物体に対して画像を取得する必要があります。たとえば、実際の生産現場では、供給されるワークには水平方向と垂直方向があるとします。水平方向のデータのみを取得してトレーニングすれば、垂直方向のワークの分類効果を保証できません。なので、データを取得するときは、**実際生産の様々なシーンを含める** 必要があります。詳細は以下に示します。

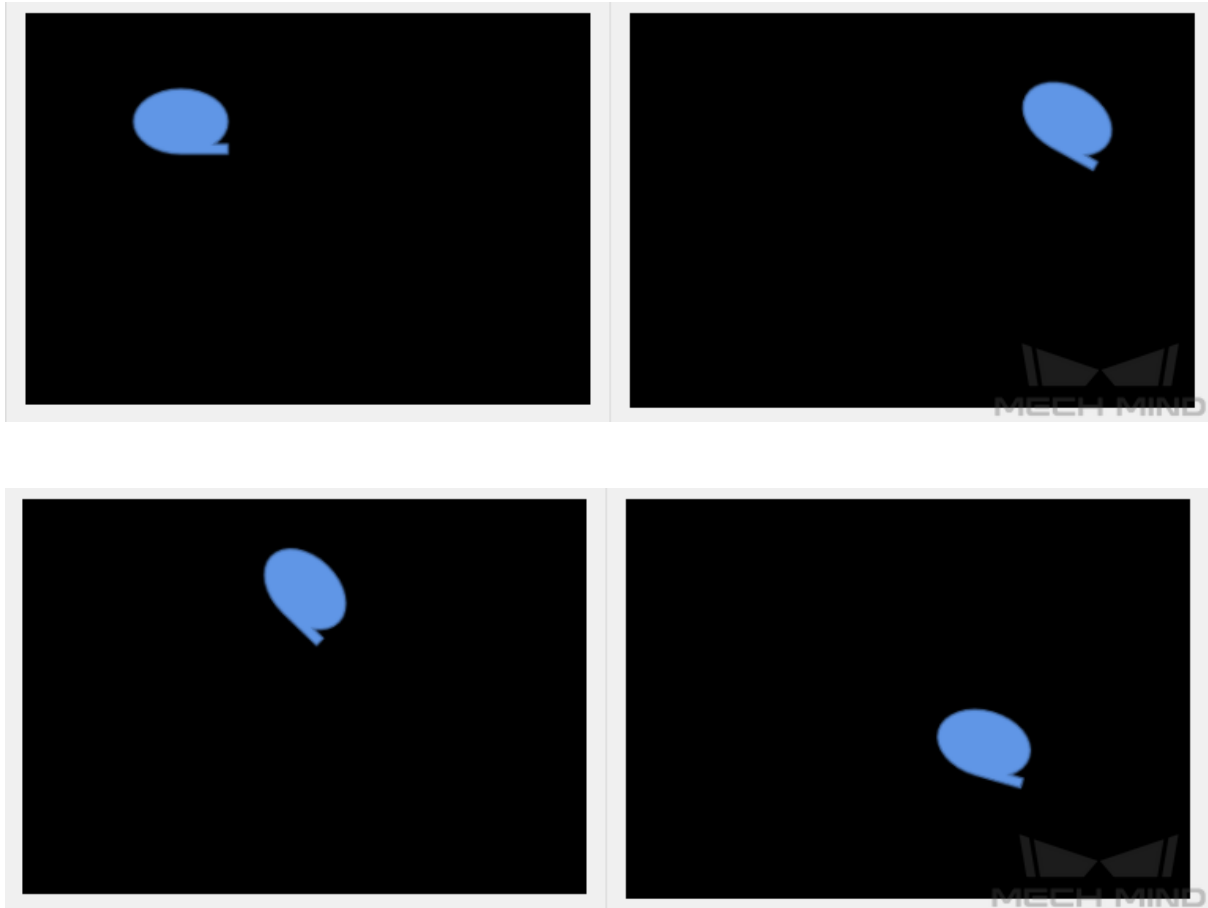
- 取得した画像データに、実際の応用シーンにありうる対象物の **異なる角度** が含まれるようにしてください。
- 取得した画像データに、実際の応用シーンにありうる対象物の **異なる位置** が含まれるようにしてください。

1. 異なる角度



異なる角度

2. 異なる位置



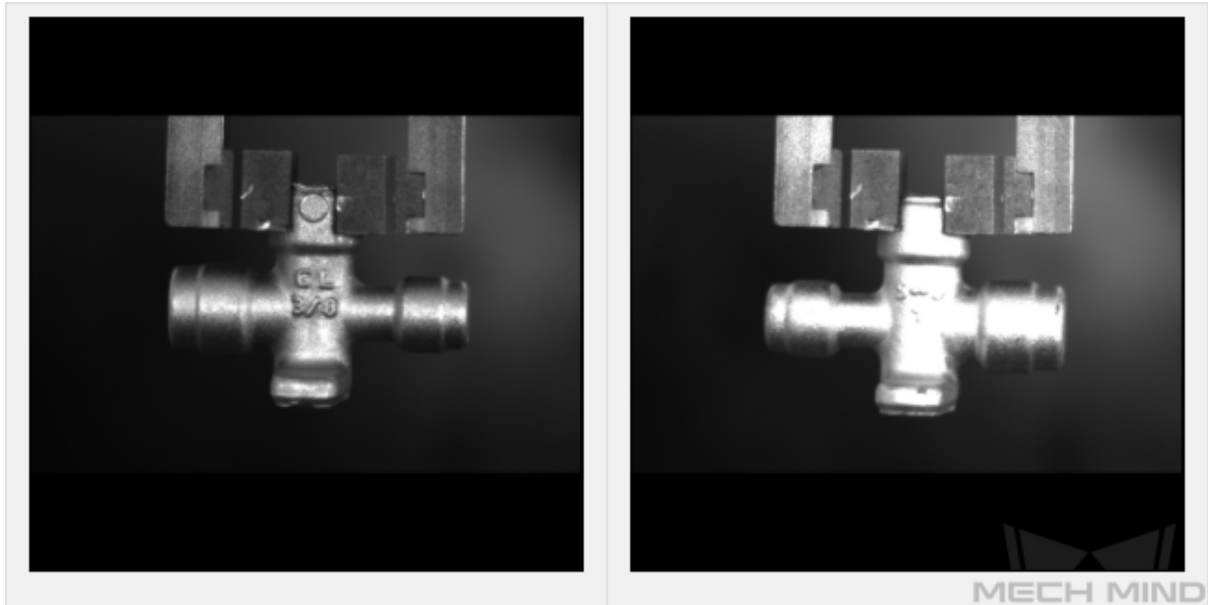
異なる位置

Mech-Vision を使用してデータを取得する

上記の取得環境、取得数、取得時の物体姿勢を確認した上で、*Mech-Vision* を使用してデータを取得する をご参照ください。

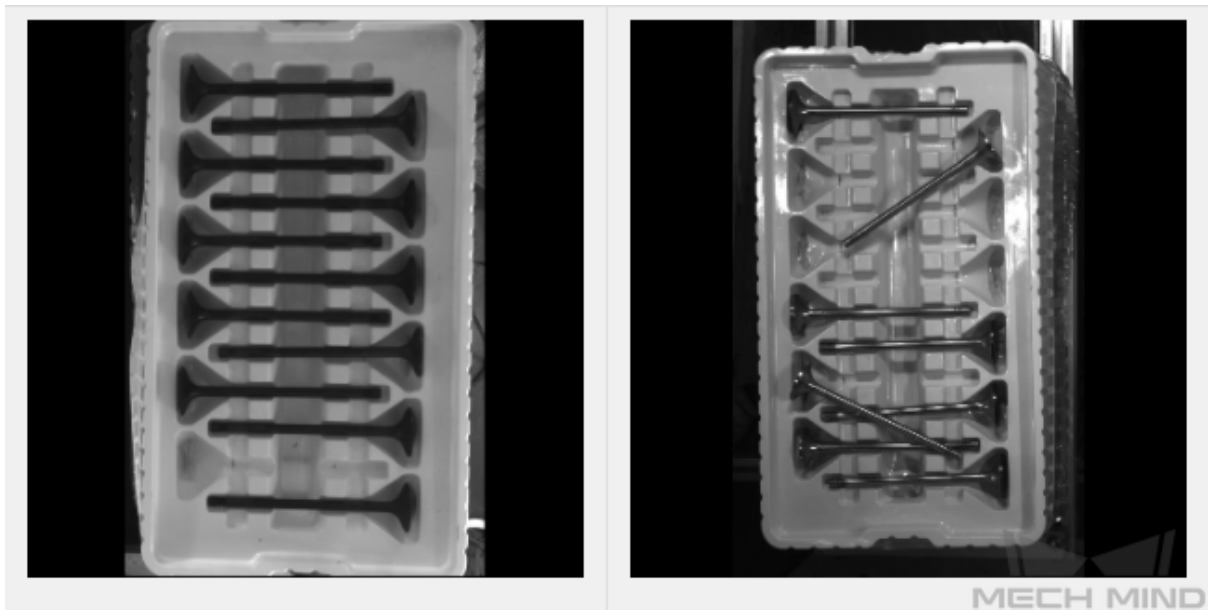
プロジェクトにおけるデータ取得の実例

1. あるバルブチューブのプロジェクトでは、単一種類の物体を認識します。バルブチューブの位置移動が小さく、正面と背面を区別する必要があり、それぞれ 10 枚を取得します。



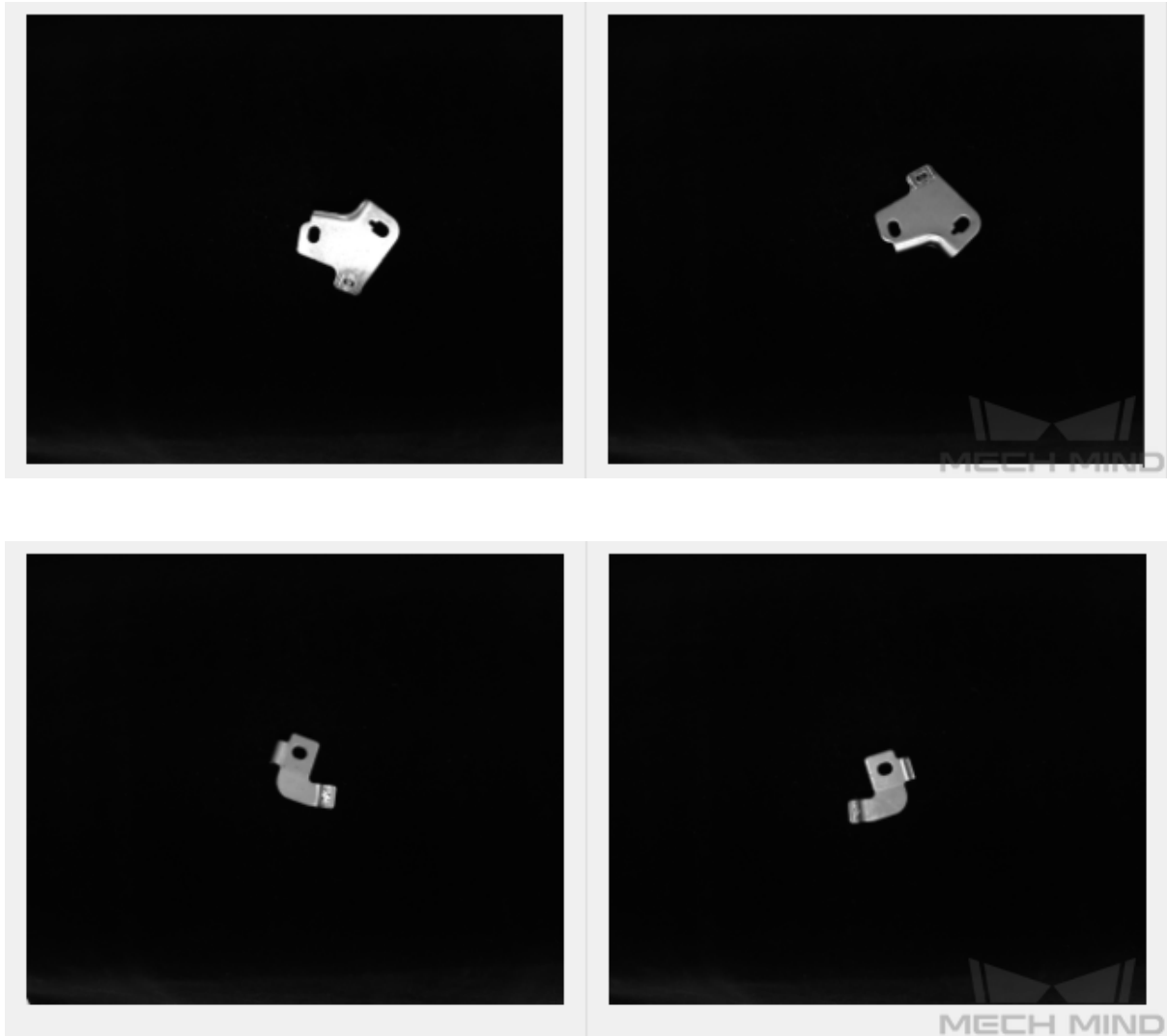
前面 / 背面

- あるエンジンバルブアセンブリのプロジェクトでは、単一種類の物体を認識します。ワークがスロット外に配置されれば、様々な姿勢がありうるので、約 20 枚の画像を取得します。スロット内に正確に配置された場合はワークの配置位置が異なる画像を約 10 枚取得します。



スロット内 / スロット外

- ある金属フレークのプロジェクトでは、2種類の物体を認識します。サイズと位置、角度を区別する必要があるため、正面と背面の画像をそれぞれ 20 枚取得します。



前面 / 背面

2.2.3 データラベリング

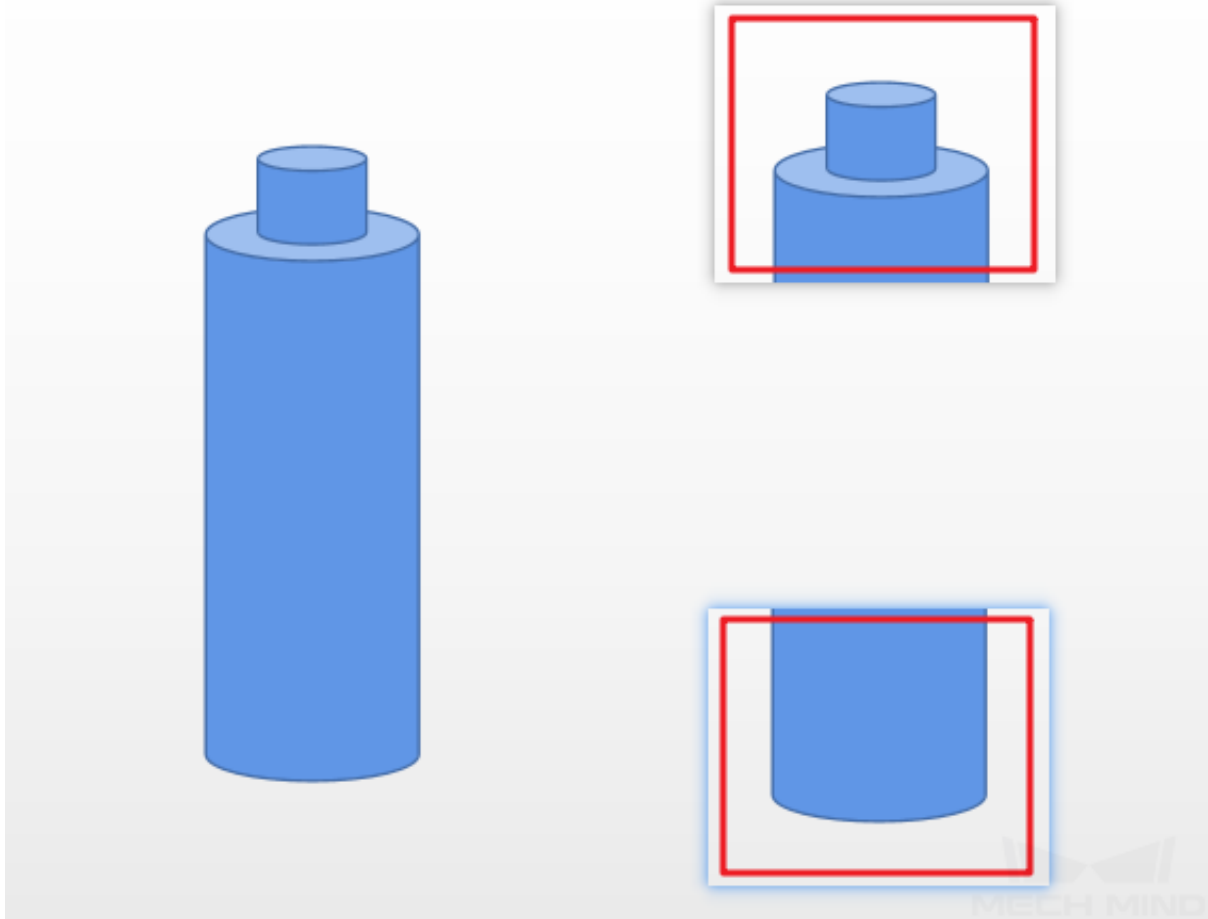
ラベルを作成する

プロジェクトのニーズに応じてラベルを作成します。たとえば、ワークの正面と背面を判別する必要がある場合、ラベル「front」と「back」を作成します。

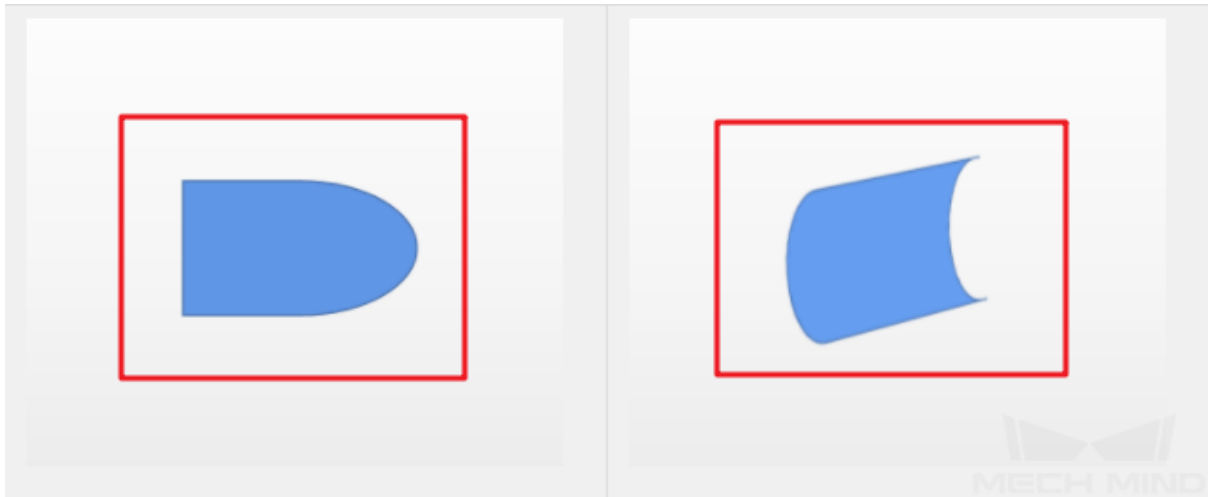
注意: ラベル名は認識しやすいものにしてください。a、b、tmpなどの意味のない名前は使用しないでください。ラベル名には英語の文字と数字以外の文字を使用しないでください。あいまいさを避けるために、日本語のローマ字（hako など）ではなく、英語の単語（box など）を使用してください。

ラベリング方法を確認する

1. 同じ物体の特徴を認識する場合は、下図に示すように、長方形のラベリングツールで明らかな特性をマークします。



2. 複数の物体カテゴリを区別する場合は、下図に示すように、物体全体を長方形のラベリングツールでマークします。



3. 実際の応用で背景が削除された画像を画像分類モデルに入力した場合、元の画像に対してラベリングを行えば、下図に示すように、物体全体の輪郭をマークします。



注意: ラベリングの品質を確認：画像分類はカテゴリの区別を目的とするため、画像とそのラベルが正確に対応していることを確認してください。例えば、10枚の「front」画像の中に1枚が「back」と誤ってラベリングされたら、モデルの効果に重大な影響を及ぼします。

2.2.4 モデルトレーニング

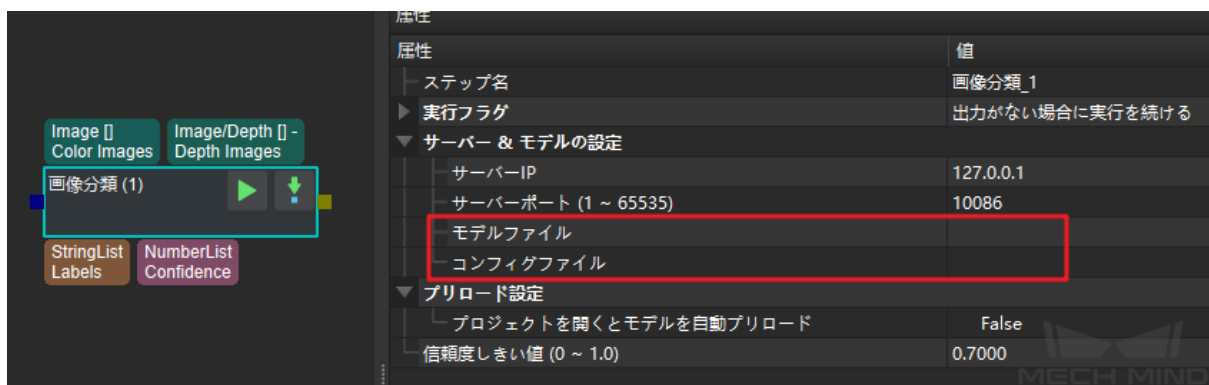
Mech-DLK を使用してモデルをトレーニングする

詳しくは、[クイックスタート](#) をご参照ください。

2.2.5 モデルを使用して予測する

Mech-Vision で画像分類モデルを使用する

注意: モデルファイルは.pth 形式で、コンフィグファイルは.json 形式です（画像分類に入力するデータは、トレーニングデータと一致させてください）。



The screenshot shows the Mech-DLK interface. On the left, a workflow is visible with nodes for 'Image [Color Images]', 'Image/Depth [Depth Images]', '画像分類 (1)', 'StringList Labels', and 'NumberList Confidence'. On the right, a configuration table is displayed with the following properties and values:

属性	値
ステップ名	画像分類_1
実行フラグ	出力がない場合に実行を続ける
▼ サーバー & モデルの設定	
サーバーIP	127.0.0.1
サーバーポート (1 ~ 65535)	10086
モデルファイル	
コンフィグファイル	
▼ プリロード設定	
プロジェクトを開くとモデルを自動プリロード	False
信頼度しきい値 (0 ~ 1.0)	0.7000

ディープラーニングを代表的なプロジェクトに適用する

3.1 紙箱のデパレタイジング・パレタイジングのプロジェクト

- 紙箱のデパレタイジング・パレタイジングのプロジェクトでは、通常、**インスタンスセグメンテーション**を使用して画像内の各紙箱を認識し、それぞれの位置情報を表示します。
- Mech-Mind 株式会社は、紙箱のデパレタイジング・パレタイジング向けに「**スーパーモデル**」を提供しています。それを Mech-Vision に直接に応用して、ほとんどの紙箱をトレーニングせずに正確な分割を実行できます。

全体的なプロセスは以下の通りです。

1. スーパーモデルを使用してプロジェクトの紙箱を全部正確に分割できるかどうかを確認します。できれば、次のステップをスキップしてスーパーモデルを使用してください。
2. できなければ、スーパーモデルで正確に分割できない全ての紙箱データを取得してトレーニングに使用します。
3. データの背景を除去します。
5. モデルをトレーニングします。
4. データをラベリングします。
6. 新しいモデルを使用して予測します。
7. プロジェクトの初期段階に各種類の紙箱を取得できない場合があります。新しい種類の紙箱を正確に分割できない場合は、ステップ 2~6 を繰り返してモデルを再更新します。

1. スーパーモデルを使用し、要件を満たすかどうかを確認する

スーパーモデルを使用してプロジェクトのデータを予測します。詳しくは、**インスタンスセグメンテーション：モデルを使用して予測**をご参照ください。

プロジェクトで使用される全ての紙箱を正確に分割できるかどうかを確認します。できれば、スーパーモデルを使用して次のステップをスキップしてください。できなければ、次のステップを実行してください。

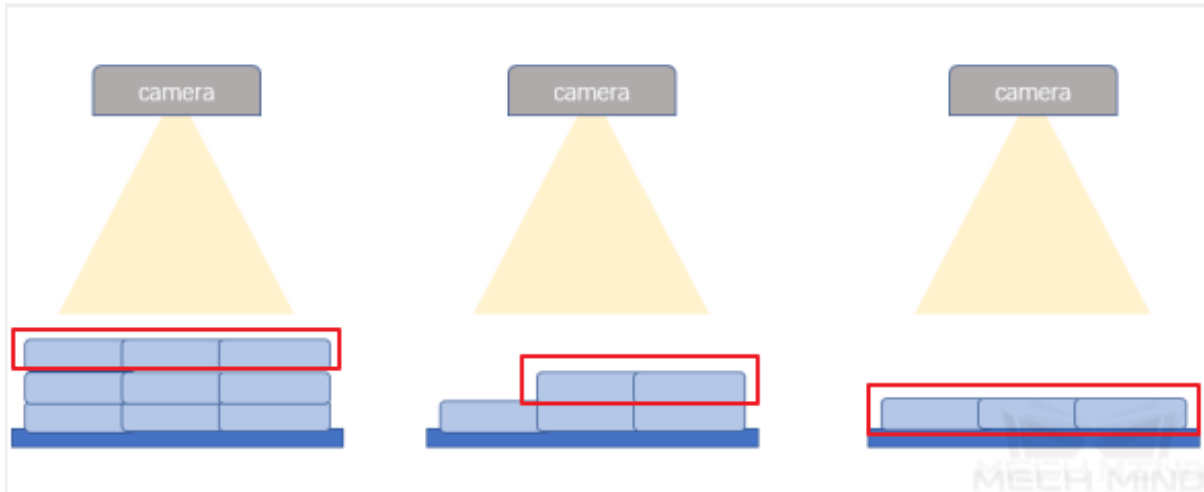
注意： スーパーモデルの効果を問わず、後のテストのためにすべてのテストデータを保留してください。

2. スーパーモデルで分割が誤った紙箱データを取得して、トレーニングに使用する

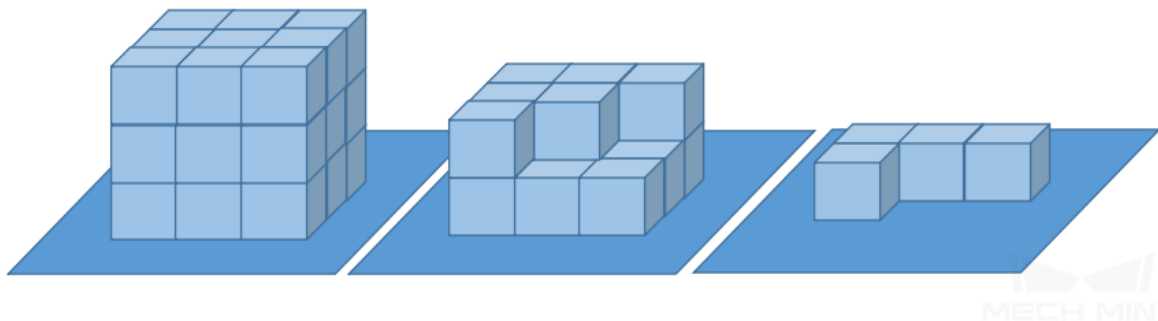
通常、スーパーモデルではほとんどの紙箱を認識できます。紙箱が密に詰め込まれたり、表面に複雑なパターンがついたりするなど、誤った分割や不完全なマスクが発生する可能性があります。ユーザーは、正確に認識できない箱に対してモデルを反復すればいいです。

たとえば、20種類の紙箱を認識するとします。その内に18種類は正確に分割できますが、2種類は分割できません。それでその2種類の紙箱のデータを取得すればいいです。また、個別に配置されれば正確に分割できますが、密に置かれれば正確に分割できない場合、密に置かれた紙箱のデータを取得すればいいです。

- 取得枚数：紙箱の種類（または配置方法）ごとに20枚の画像を取得します。
- データの要件：①異なるパレットパターンの各層（高層、中層、最下層）に密に配置された紙箱の画像を10枚取得します。②異なるパレットパターンの各層（高層、中層、最下層）のフルパレット、ハーフパレット、少量の紙箱の画像を10枚取得します。



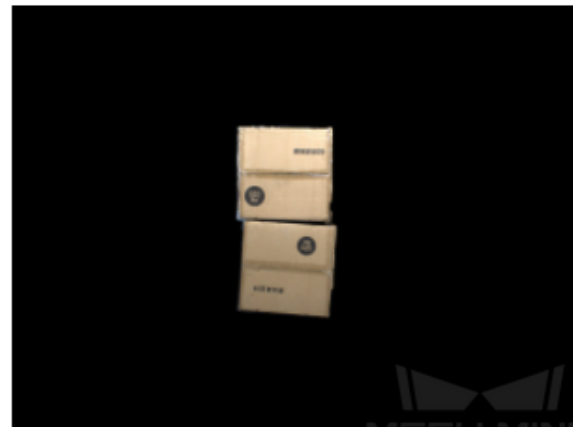
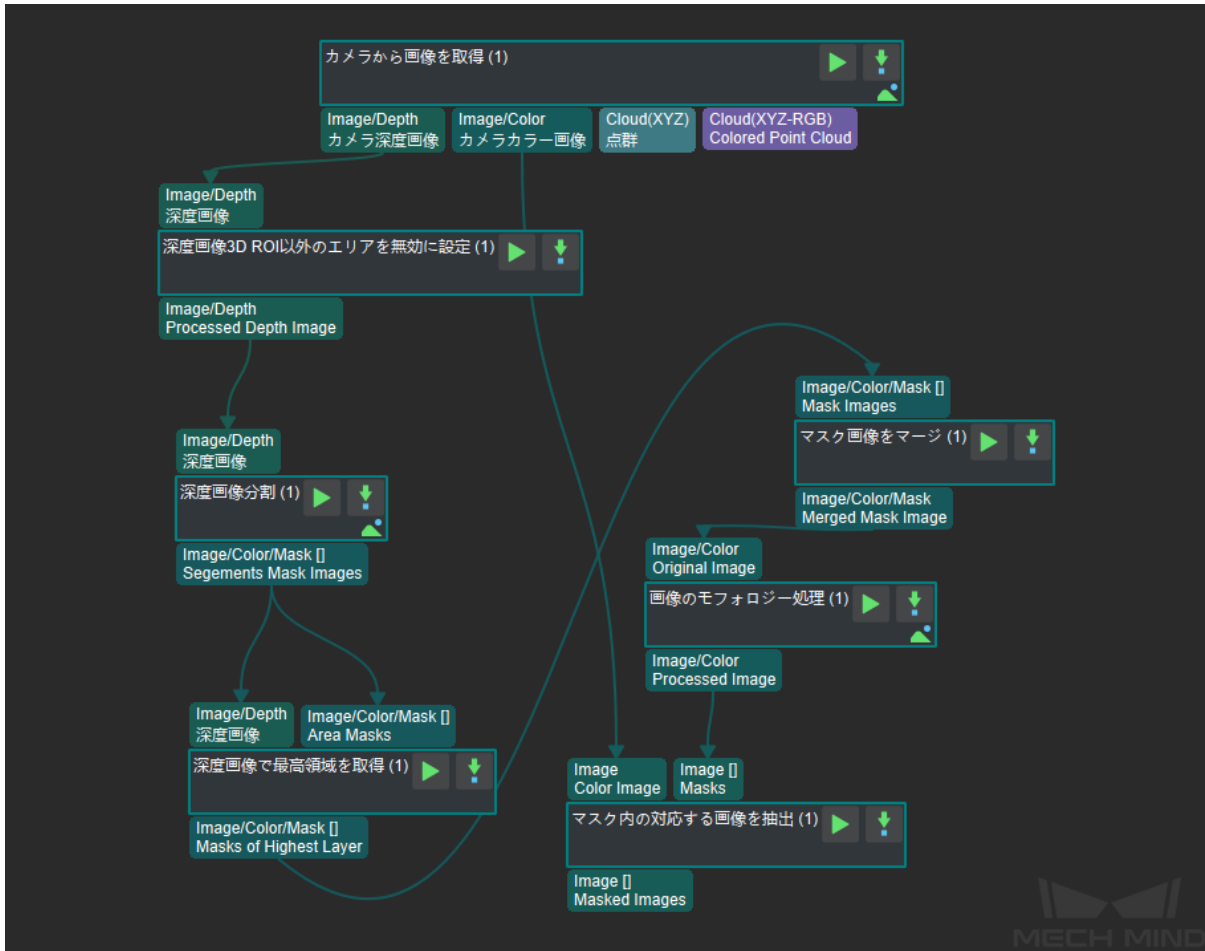
高層（左）/中層（中）/最下層（右）



最高層のフルパレット（左）/中層のハーフパレット（中）/最下層の少量パレット（右）

3. データの背景を除去する

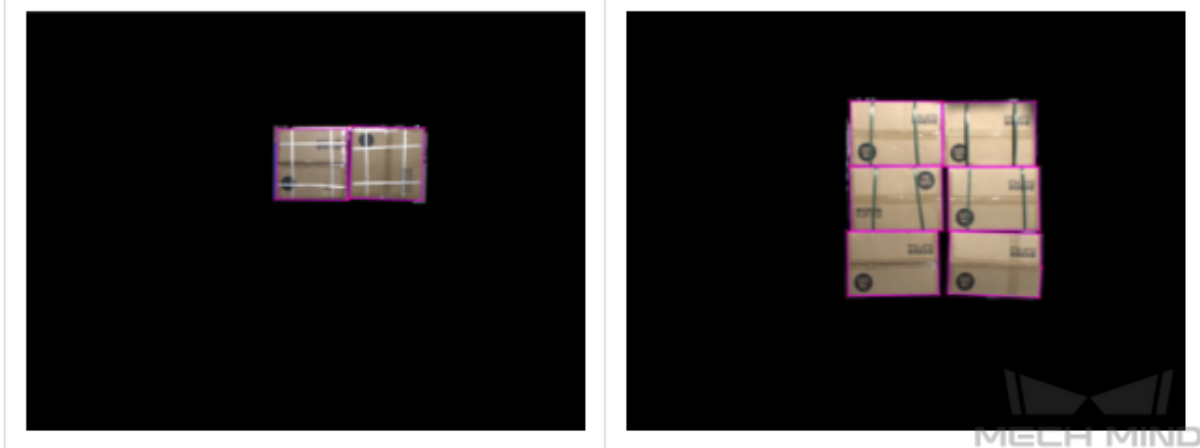
紙箱のデパレタイジング・パレタイジングのシーンでは、紙箱は全てパレットに積み上げられるので、背景除去により干渉を回避し、モデルの効果を大幅に改善できます。Mech-Visionでは、背景除去のステップの組合せにより背景除去データを取得します。



元の画像 / 背景除去後の画像

4. データをラベリングする

詳しくは、[インスタンスセグメンテーション：データラベリング](#) をご参照ください。
 紙箱のデパレタイジング・パレタイジングのシーンでは、紙箱の上面のみをマークし、側面はマークしません。また、完全な上面のみをマークし、遮蔽された箱をマークしません。



上面の輪郭をマーク

5. モデルをトレーニングする

詳しくは、[インスタンスセグメンテーション：モデルトレーニング](#)をご参照ください。

トレーニングのパラメータについて、エポックの合計数（epoch）を「200」に設定し、他のパラメータは初期値を使用します。

6. 新しいモデルを使用して予測する

詳しくは、[インスタンスセグメンテーション：モデルを使用して予測](#)をご参照ください。

7. 必要に応じて、ステップ 2-6 を繰り返してモデルを再更新する

多種類の紙箱がある場合、プロジェクトの初期段階では各種類の紙箱を取得してテストすることはできません。この場合、モデルを使用して既存の種類紙箱をテストします。新しい種類の紙箱を正確に分割できない場合は、ステップ 2-6 を繰り返して、モデルを再更新します。

注釈：スーパーモデル：大規模なデータトレーニングに基づく特定の種類の物体（紙箱、麻袋、宅配便小包など）に適した一般的なディープラーニングモデルを指します。

3.2 麻袋のデパレタイジング・パレタイジングのプロジェクト

- 麻袋のデパレタイジング・パレタイジングのプロジェクトでは、通常、[インスタンスセグメンテーション](#)を使用して画像内の各麻袋を分割します。
- Mech-Mind 株式会社は、麻袋のデパレタイジング・パレタイジング向けに「スーパーモデル」を提供しています。それを Mech-Vision に直接に應用して、ほとんどの麻袋をトレーニングせずに正確な分割を実行できます。

全体的なプロセスは以下の通りです。

1. スーパーモデルを使用してプロジェクトの麻袋を全部正確に分割できるかどうかを確認します。できれば、次のステップをスキップしてスーパーモデルを使用してください。
2. できなければ、スーパーモデルで正確に分割できない全ての紙箱データを取得してトレーニングに使用します。
3. データをラベリングします。

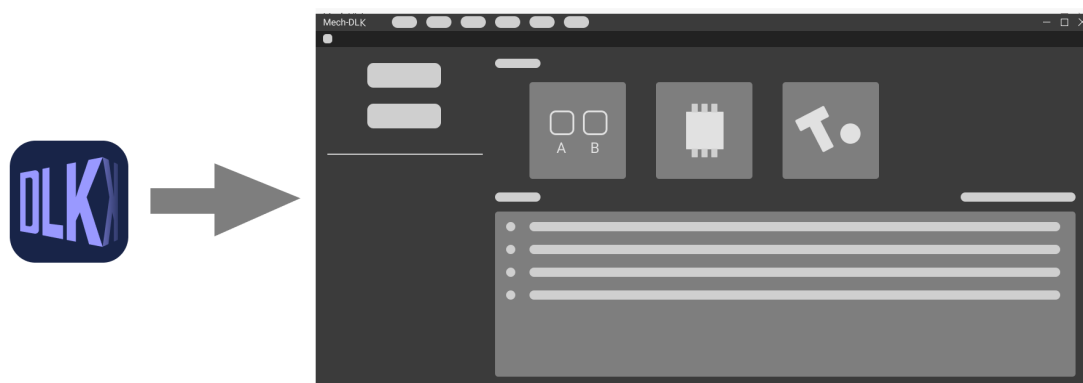
4. モデルをトレーニングします。
5. 新しいモデルを使用して予測します。
6. 多種類の麻袋がある場合、プロジェクトの初期段階では各種類の麻袋を取得してテストすることはできません。この場合、モデルを使用して既存の種類 of 麻袋をテストします。新しい種類の麻袋を正確に分割できない場合は、ステップ 2~6 を繰り返して、モデルを再更新します。

MECH-DLK 使用ガイド

4.1 ソフトウェアの紹介

4.1.1 ソフトウェアの機能

Mech-DLK は Mech-Mind 株式会社が独自に開発したプラットフォームベースのディープラーニングソフトウェアであり、業界をリードする複数の AI アルゴリズムを組み込んでおり、積み重ねられた物体の認識、複雑な欠陥検出、製品グレードの分類などの複雑な問題を直感的でシンプルな操作で解決できます。家庭電化製品、新エネルギー、自動車、家電、物流などの業界に適しており、生産効率と製品の良品率を向上させ、生産ラインの人件費を削減できます。



4.1.2 モジュールの機能

画像分類：異なる種類の対象物を分類するために使用されます。少量の多種類対象物の画像を提供するだけで、学習を通じて異なる種類の対象物を区別できます。

対象物検出：画像の対象物をすばやく位置決めするために使用されます。少量の画像を提供するだけで、学習を通じて対象物の位置を決めます。

セマンティックセグメンテーション：画像内の欠陥の位置を位置決めするために使用されます。一定量の OK と NG サンプルを提供するだけで、学習を通じて画像に欠陥があるかどうかを検出できます。

画像分類：異なる種類の対象物を位置決めし、分類するために使用されます。少量の多種類対象物の画像を提供するだけで、学習を通じて異なる種類の対象物を位置決めし、種類のラベルを出力できます。

注意: 上記のモジュールは、いずれもユーザーのニーズを満たすためにモデルを個別にトレーニングで
きます。様々な機能を統合するカスタマイズのニーズがあれば、Mech-DLK で異なるモジュールを連結
してトレーニングを組み合わせることができます。

4.1.3 カスタマイズ開発に対応する

カスタマイズ開発の構成要件	
オペレーティングシステム	Windows7 以上
開発プラットフォーム	VS2013、VS2015、VS2017、VS2019 (推奨)
開発言語	C、C++、C#

4.2 環境構成

4.2.1 ハードウェア要件

	Mech-DLK Pro-Run	Mech-DLK Pro-Train/Standard
オペレーティングシステム	Windows10	
CPU	i5 以上	i7 以上
メモリ	8GB 以上	16GB 以上
グラフィックカード	GeForce GTX 1650 (4GB) 以上	GeForce GTX 2070 (8GB) 以上

グラフィックカード型番	デスクトップ	ノートパソコン
10シリーズ	NVIDIA GeForce GT 1030 NVIDIA GeForce GTX 1050 NVIDIA GeForce GTX 1050Ti NVIDIA GeForce GTX 1060 NVIDIA GeForce GTX 1070 NVIDIA GeForce GTX 1070Ti NVIDIA GeForce GTX 1080 NVIDIA GeForce GTX 1080Ti NVIDIA GeForce GTX 1650 NVIDIA GeForce GTX 1650 SUPER NVIDIA GeForce GTX 1660 NVIDIA GeForce GTX 1660Ti NVIDIA GeForce GTX 1660 SUPER	NVIDIA GeForce GTX 1050 NVIDIA GeForce GTX 1050Ti NVIDIA GeForce GTX 1060 NVIDIA GeForce GTX 1070 NVIDIA GeForce GTX 1080 NVIDIA GeForce GTX 1650 NVIDIA GeForce GTX 1650 NVIDIA GeForce GTX 1650Ti NVIDIA GeForce GTX 1660Ti
20シリーズ	NVIDIA GeForce RTX 2060 NVIDIA GeForce RTX 2060 SU- PER NVIDIA GeForce RTX 2070 NVIDIA GeForce RTX 2070 SUPER NVIDIA GeForce RTX 2080 NVIDIA GeForce RTX 2080Ti NVIDIA GeForce RTX 2080 SUPER	NVIDIA GeForce RTX 2060 NVIDIA GeForce RTX 2070 NVIDIA GeForce RTX 2080
30シリーズ	NVIDIA GeForce RTX 3050 NVIDIA GeForce RTX 3060 NVIDIA GeForce RTX 3060Ti NVIDIA GeForce RTX 3070 NVIDIA GeForce RTX 3070Ti NVIDIA GeForce RTX 3080 NVIDIA GeForce RTX 3080Ti NVIDIA GeForce RTX 3090	NVIDIA GeForce RTX 3050 Laptop GPU NVIDIA GeForce RTX 3060 Laptop GPU NVIDIA GeForce RTX 3070 Laptop GPU NVIDIA GeForce RTX 3080 Laptop GPU

4.2.2 環境構成

1、V2.1.0 環境をインストールする

一般的なインストール方法で、いかなるバージョンのディープラーニングをインストールしたこともない新しいコンピューターに適します。

- Mech-Mind_software_environment_installer(2.1.0).zip.001 と Mech-Mind_software_environment_installer(2.1.0).zip.002、Mech-Mind_software_environment_installer(2.1.0).zip.003 を同じパスにダウンロードします。
- ファイルのサイズが大きいため、ダウンロードしたあと、ファイルをチェックしてください。チェックするには、ファイルを右クリックし、CRC SHA -> CRC-32 を選択してチェックサムを確認します。上記の三つのファイルの標準チェックサムはそれぞれ下記の通りです。

Name **Mech-Mind_software_environment_installer(2.1.0).zip.001**

Size **4290772992** Bytes (4092 MiB)

CRC32 **903E692A**

Name **Mech-Mind_software_environment_installer(2.1.0).zip.002**

Size **4290772992** Bytes (4092 MiB)

CRC32 **C95B7D3F**

Name **Mech-Mind_software_environment_installer(2.1.0).zip.003**


Size **918773062** Bytes (876 MiB)


CRC32 **C36F7F64**




- 解凍ソフトを使用して上記の圧縮ファイルを解凍します。
- 解凍した Mech_Mind_software_environment_installer.exe をダブルクリックして画面の指示に従ってインストールします。

▶ **Mech-Mind_software_environment_installer(2.1.0)**

 **required_resource_packs**

 **Mech_Mind_software_environment_installer**

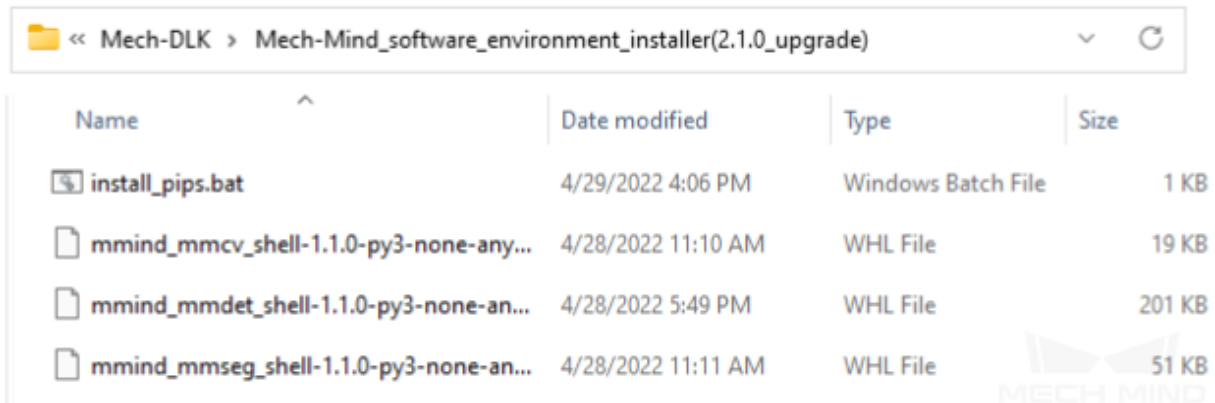
 **readme**



2、V2.0.0 環境から V2.1.0 環境にアップデートする

現在 V2.0.0 バージョンの環境をインストールしたコンピューターに適します。すなわち、Mech-Mind_software_environment_installer(V2.0.0) 又は Mech-Mind_software_environment_installer(V2.0.0_Lite) 環境のインストールパッケージをインストールしたコンピューターに適します。

- Mech-Mind_software_environment_installer(2.1.0_upgrade) フォルダをダウンロードします。
- ダウンロードした後、install_pips.bat をダブルクリックして環境のアップデートを実行します。



3、V1.4.0 又はより古い環境から V2.1.0 環境にアップデートする

現在、V1.4.0 バージョン又はより古いバージョンの環境をインストールしたコンピューターに適します。すなわち、既に Mech-Mind_software_environment_installer(V1.4.0) 又はより古い環境のインストールパッケージをインストールし、V2.0.0 バージョンの環境のインストールパッケージをインストールしていないコンピューターに適します。

- Mech-DLK トレーニングモデルを使用する場合に、上記の「1、V2.1.0 環境をインストールする」をご参照ください。
- Mech-DLK トレーニングモデルを使用する場合
 - Mech-Mind_software_environment_installer(2.1.0_lite) フォルダをダウンロードします。
 - ダウンロードした後、Mech_Mind_software_environment_installer.exe をダブルクリックして画面の指示に従ってインストールします。

Mech-Mind_software_environment_installer(2.1.0_lite)

required_resource_packs

Mech_Mind_software_environment_installer

readme

4. インストールガイド

現在の環境のバージョン	推奨操作	ダウンロードするファイル	ダウンロード後の操作	注
いかなるバージョンをインストールしたこともない	完全なV2.1.0環境をインストールする	Mech-Mind_software_environment_installer(2.1.0).zip.001 Mech-Mind_software_environment_installer(2.1.0).zip.002 Mech-Mind_software_environment_installer(2.1.0).zip.003	<ul style="list-style-type: none"> 解凍ソフトを使用してファイルを解凍する 解凍したexe形式のインストールプログラムを実行する 	<ul style="list-style-type: none"> ダウンロードした三つのファイルを同じパスに置いてください。 ダウンロードした後、ファイルをチェックしてください。
V2.0.0	V2.1.0_upgradeをインストールする	Mech-Mind_software_environment_installer(2.1.0_upgrade)	<ul style="list-style-type: none"> install_pips.batを実行する 	
V1.4.0と以下	Mech-DLKトレーニングモデルを使用する場合	Mech-Mind_software_environment_installer(2.1.0).zip.001 Mech-Mind_software_environment_installer(2.1.0).zip.002 Mech-Mind_software_environment_installer(2.1.0).zip.003	<ul style="list-style-type: none"> 解凍ソフトを使用してファイルを解凍する 解凍したexe形式のインストールプログラムを実行する 	<ul style="list-style-type: none"> ダウンロードした三つのファイルを同じパスに置いてください。 ダウンロードした後、ファイルをチェックしてください。
	Mech-DLKトレーニングモデルを使用しない場合	Mech-Mind_software_environment_installer(2.1.0_lite)	<ul style="list-style-type: none"> フォルダーにあるexe形式のインストールプログラムを実行する 	

現在の環境のバージョンを確認する方法

コントロールパネルを開き、プログラムと機能で現在の環境のバージョンを確認できます。

WPS Office (11.1.0.9192)	Kingsoft Corp.	2021/2/22	590 MB	11.1.0.9192
CLMake	Kiware	2022/3/8	107 MB	3.22.3
Netron 5.6.6	Lutz Roeder	2022/3/26	420 MB	5.6.6
Mech-Vision	Mech-Mind	2022/4/12	1.71 GB	1.5.1
Mech-Viz	Mech-Mind	2022/4/8	3.37 GB	1.5.1-dev-04-07
Mech-Center	Mech-Mind	2022/4/6	319 MB	1.5.1
Mech-Eye SDK	Mech-Mind	2022/5/5	824 MB	1.5.2
Mech-Viz	Mech-Mind	2022/4/12	3.37 GB	1.5.1-dev-04-11
Mech-DLK	Mech-Mind Robotics Technologies Ltd	2022/4/28	2.67 GB	2.0.0
Mech Mind Software Environment	Mech-Mind Robotics Technologies Ltd	2022/4/28	2.0.0	2.0.0
Microsoft Visual Studio Installer	Microsoft Corporation	2021/6/29	3.12 GB	8.9.9.1
Microsoft OneDrive	Microsoft Corporation	2022/5/5	238 MB	22.077.0410.0007
Microsoft Visual Studio Code (1.66.1)	Microsoft Corporation	2022/4/11	282 MB	1.66.1

4.3 クイックスタート

- 本節では、モデルをすばやくトレーニングしてエクスポートする方法について説明します。このモデルは、対象物の欠陥を識別し、欠陥の種類を分類することができます。
- Mech-DLK を使用することで、ユーザーは画像分類モジュールをセマンティックセグメンテーションモジュールとカスケード接続することでこの機能を実現できます。
- Mech-DLK を使用する前に、**ディープラーニング環境** および Mech-DLK が成功にインストールされたかどうかを確認してください。

1. プロジェクトを新規作成する

図1に示すように、ホームページの **プロジェクトを新規作成** をクリックし、新しいプロジェクトを作成します。

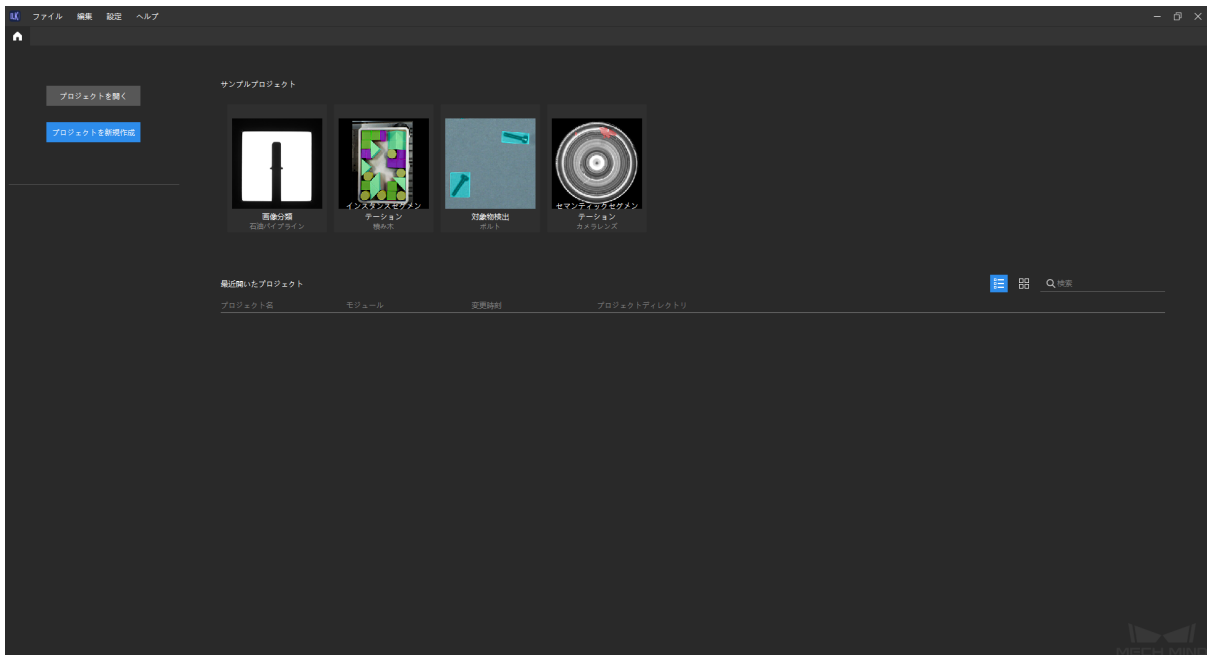



図1 プロジェクトを新規作成する

2. モジュールを選択する

モジュールを追加：インターフェースの右側にあるモジュールバーの  をクリックして「セマンティックセグメンテーション」を選択し、「確認」をクリックします。

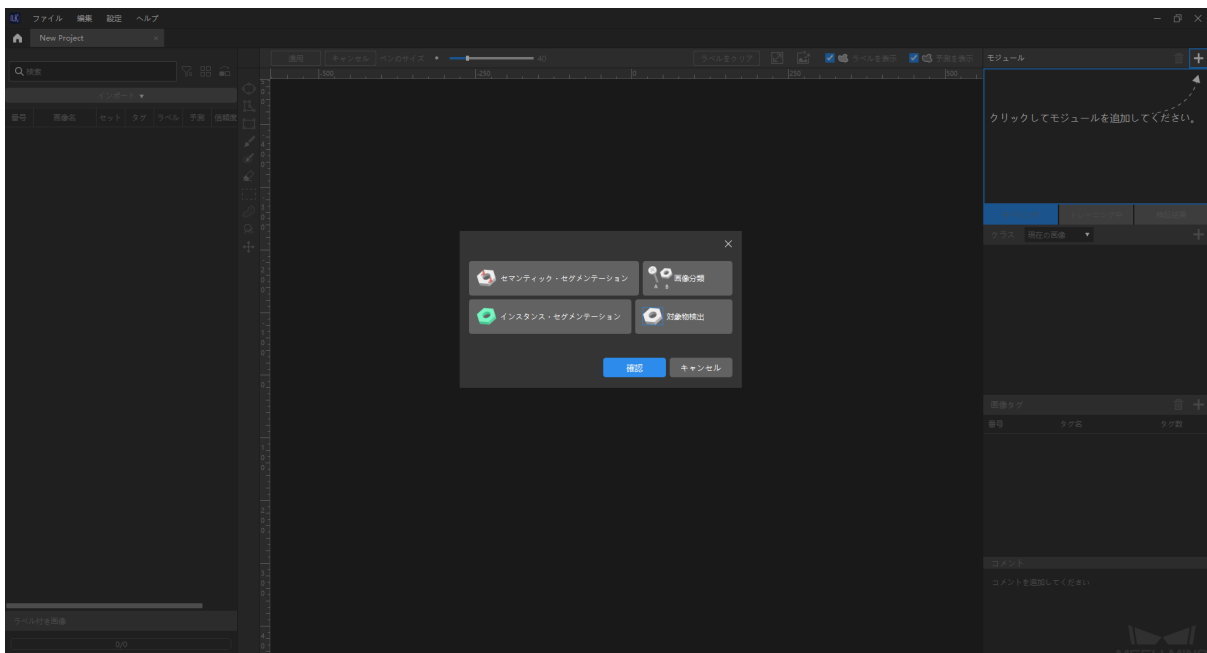


図2 モジュールを追加する

3. データをインポートする

左上隅にある **インポート** をクリックして、インポートするローカル画像を選択します。

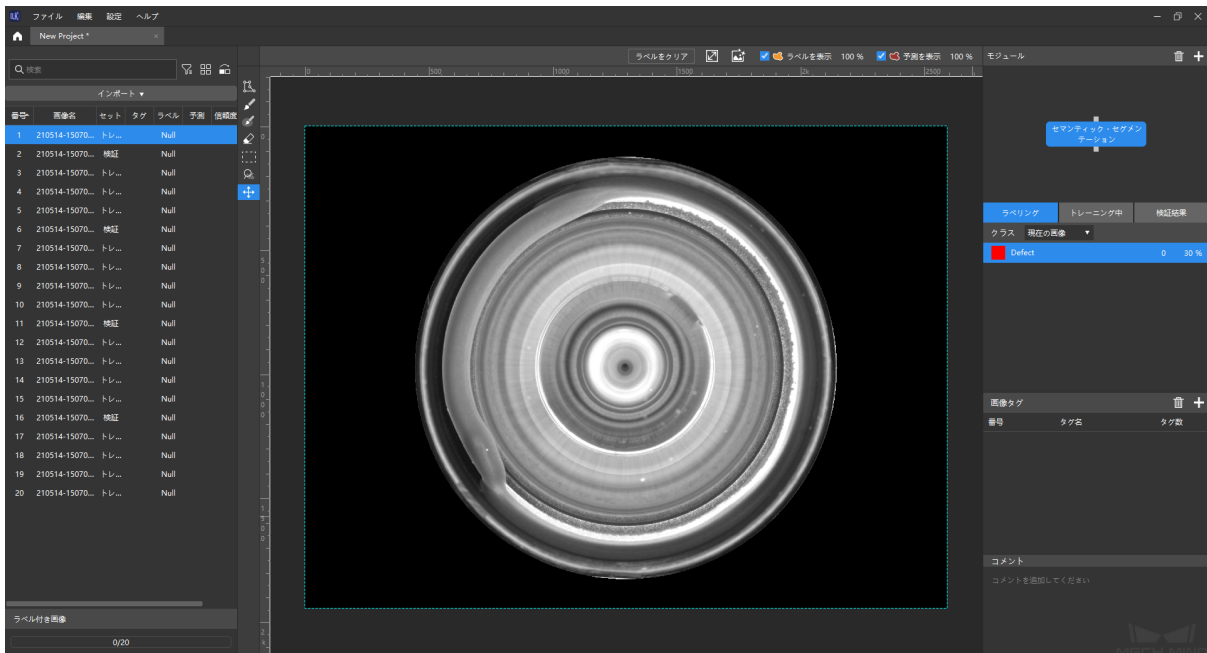



図3 データをインポートする

注意: セマンティックセグメンテーションモジュールによってインポートされた画像には、欠陥のない画像が含まれている必要があり、検証セットとトレーニングセットにもそれが含まれている必要があります。それがないと、ポップアップウィンドウが表示され、後のトレーニングを正常に実行できません。

4. データをラベリングする

セマンティックセグメンテーションは、「Defect」というラベルを自動的に生成します。データをラベリングするには、左側のツールバー  から適切なツールを選択してください。

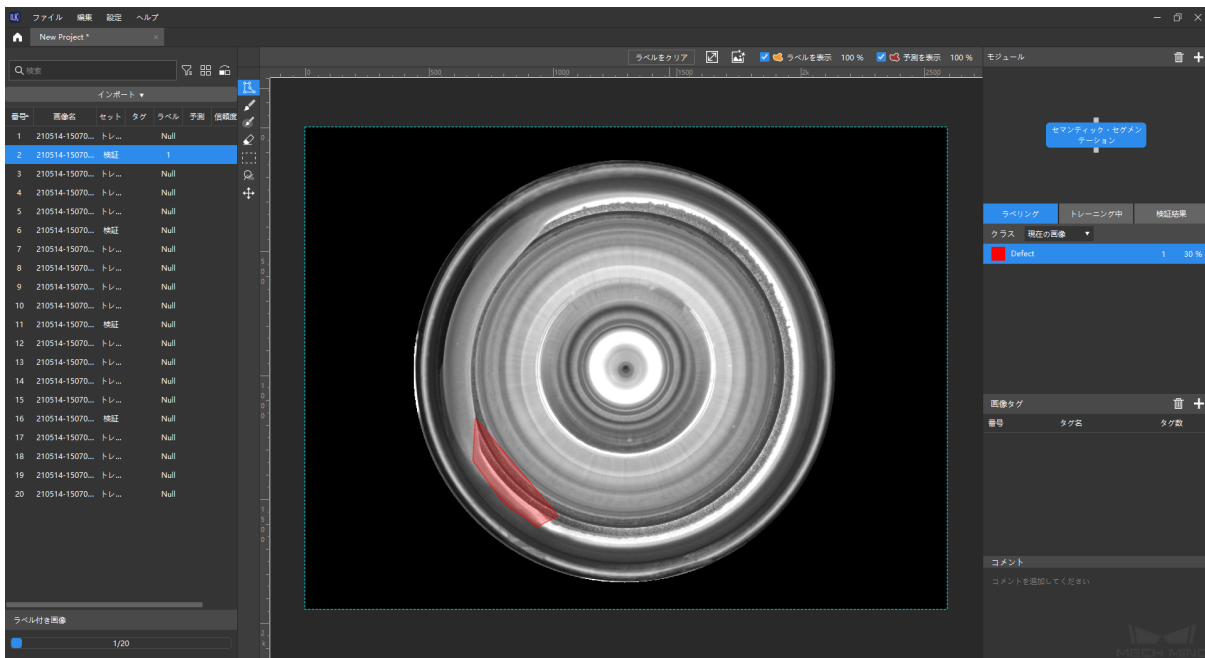


図4 セマンティックセグメンテーション・ラベリング

5. モデルをトレーニングする

右下隅にある **トレーニング** をクリックして、トレーニングを開始します。**チャートを表示** をクリックすれば、トレーニングの正確度とロスが表示されます。

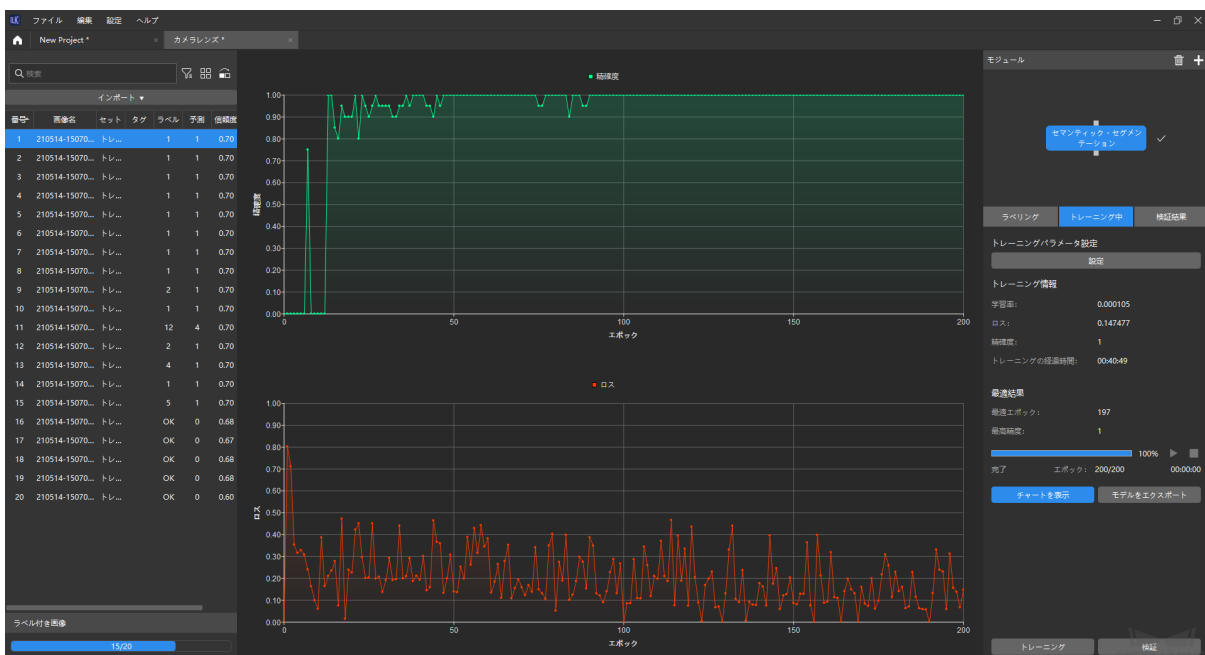


図5 トレーニングチャート

6. モデルを検証する

セマンティックセグメンテーションモデルがトレーニングされたら、**検証** をクリックして結果を検証します。検証した後、トレーニング結果が保存されます。

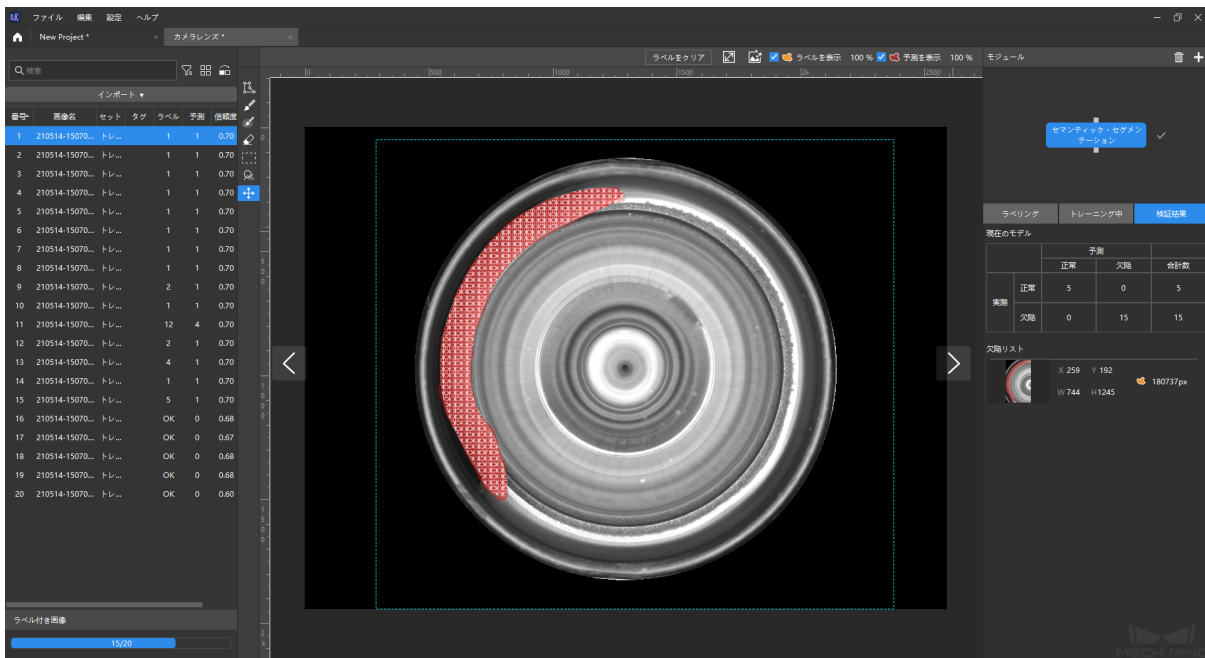



図6 検証チャート

7. 次のモジュールを追加する

モデルの効果が良好であることを確認したら、ページの右上隅にあるモジュールバーの  をクリックして、画像分類アルゴリズムを追加します。

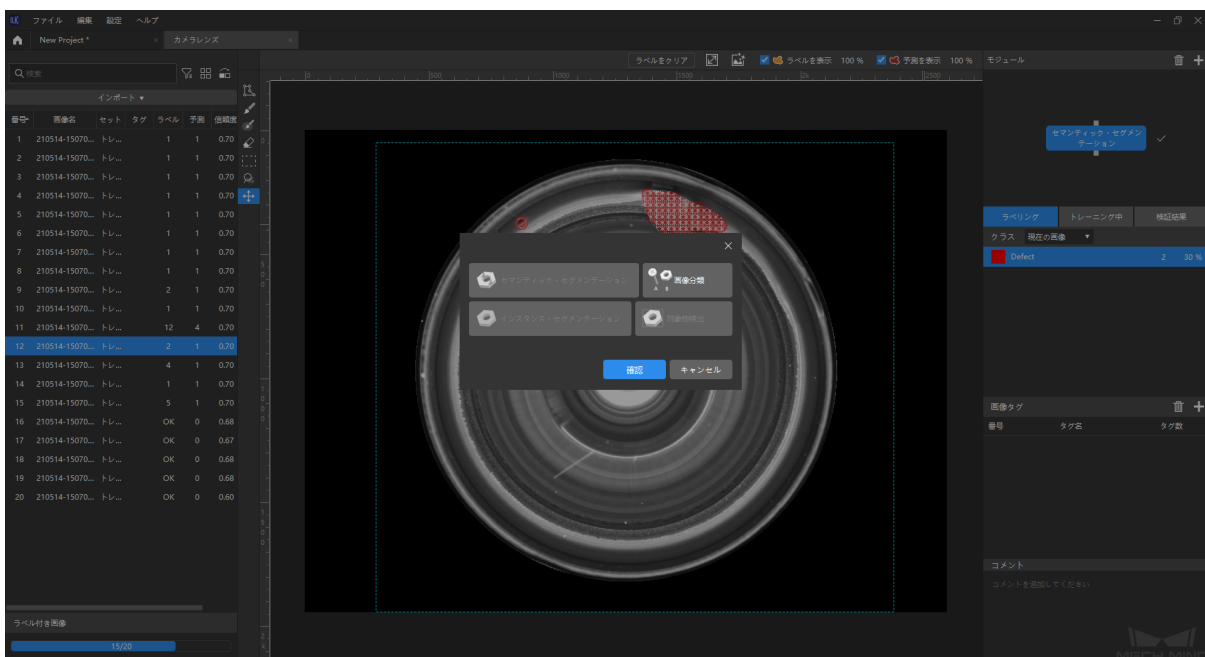


図7 アルゴリズムを追加する

8. データを「画像分類」モジュールにインポートする

前のモデルのトレーニング結果は、データソースとして画像分類アルゴリズムにインポートされます。

「インポート」をクリックすると、選択可能なデータがポップアップボックスに表示されます。このとき、必要なデータを手動で選択してください。

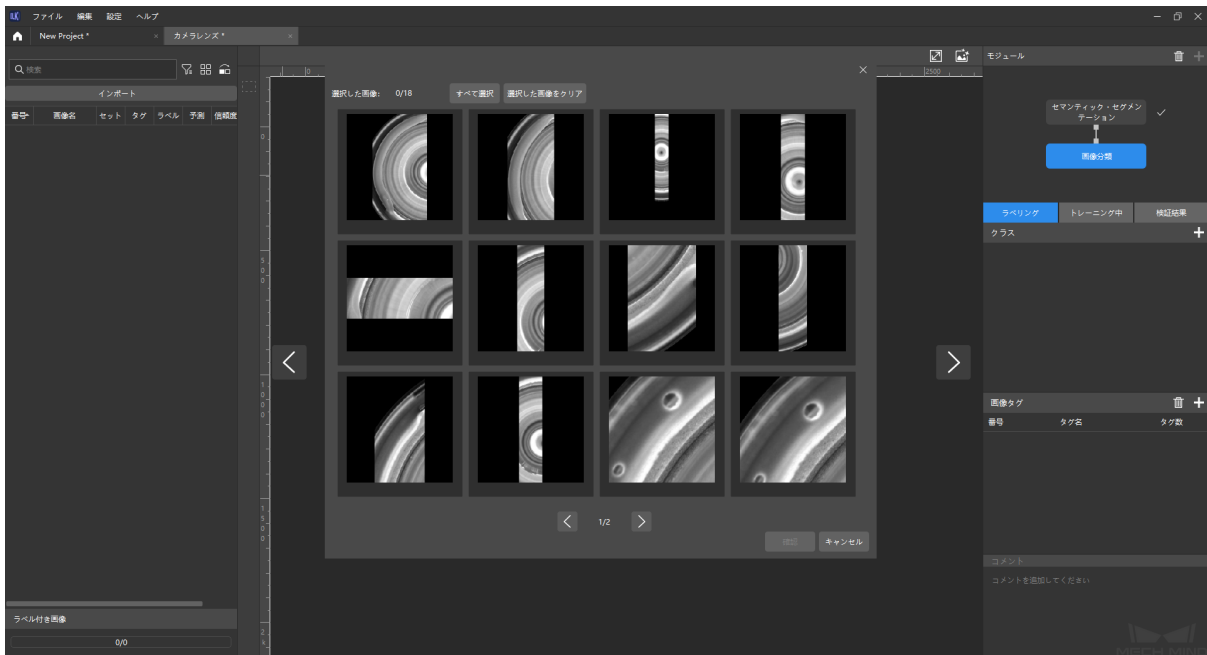





図 8 データをインポートする

9. 画像分類ラベルを作成してラベリングする

画像分類はラベリングの前に、左下隅のラベルバーの  をクリックして、異なる種類の対象物にラベルを作成する必要があります。次に、ラベルバーの左側にある   をクリックして、画像にラベルを付けます。

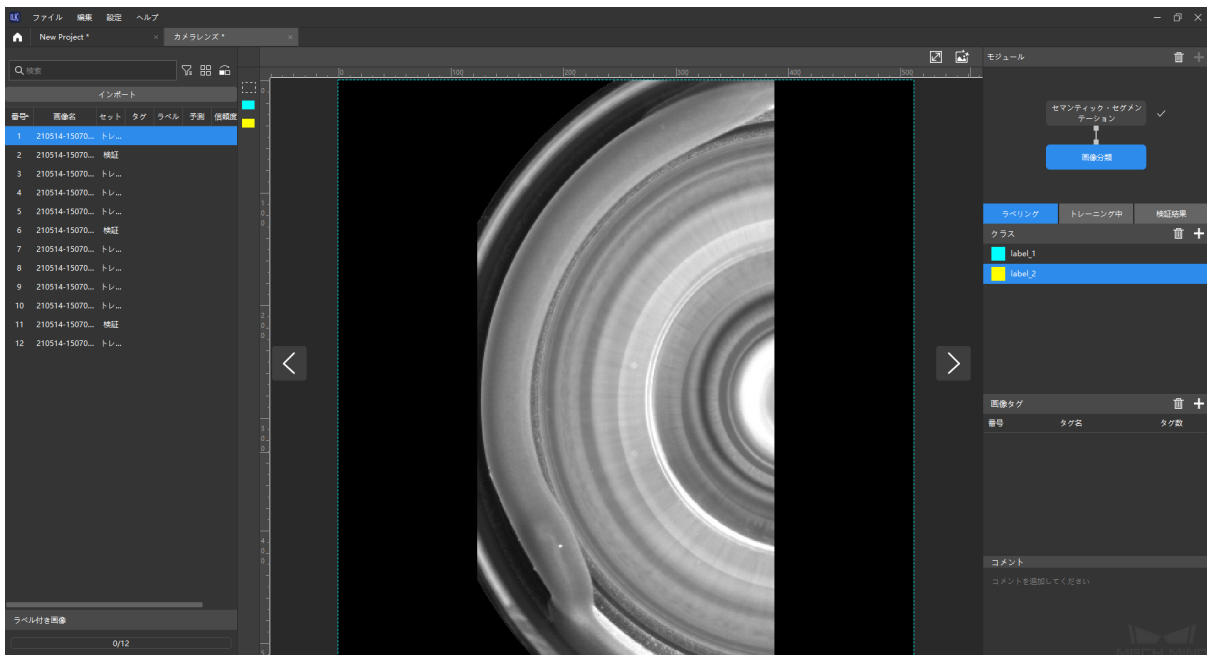


図 9 画像分類ラベリング

10. モデルをトレーニングする

右下隅にある **トレーニング** をクリックして、トレーニングを開始します。**チャートを表示** をクリックすれば、トレーニングの正確度とロスが表示されます。

11. 最後のモデルをエクスポートする

図 10 に示すように、モデルのトレーニングが終了したら、右側にある **モデルをエクスポート** をクリックし、保存パスを選択して最適モデルをフォルダにエクスポートします。

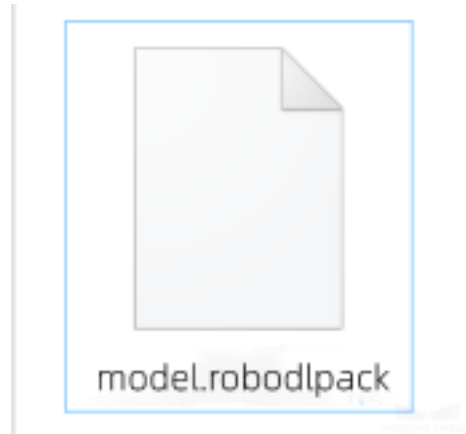


図 10 モデルファイル

4.4 専門用語

ラベリング：画像内の対象物を手動で選択します。

データセット：Mech-DLK によってエクスポートされた dlkdb 形式のラベル情報を含むファイルです。

ラベル付き：手動でラベルを付けた画像データです。

ラベルなし：ラベルのない画像データです。

トレーニングセット：モデルトレーニングに使用される、手動でラベルを付けた画像データ画像のデータセットです。

検証セット：モデル効果の検証に使用される、手動でラベルを付けた画像のデータセットです。

OK 画像：欠陥のない画像です。

NG 画像：欠陥がある画像です。

トレーニング：「トレーニングセット」を使用してディープラーニングモデルをトレーニングするプロセスです。

検証：トレーニング済みモデルを使用してデータを計算し、結果を出力します。

精確度：モデルが検証セットを予測する場合、サンプルの総数に対する正しく予測されたサンプルの数の比率です。

ロス：検証セットがモデルによって予測された結果と実際の結果との不一致の程度を推定します。

エポック：ディープラーニングアルゴリズムを利用してトレーニングデータセット全体で学習を実行した回数です。

4.5 ショートカット

番号	機能	ショートカット	説明
1	プロジェクトを新規作成	Ctrl + n	
2	プロジェクトを保存	Ctrl + s	
3	プロジェクトを開く	Ctrl + o	
4	ラベリングを元に戻す (Undo)	Ctrl + y	
5	ラベリングをやり直す ((Redo)	Ctrl + z	
6	ラベリングをコピー	Ctrl + c	
7	ラベリングを貼り付ける	Ctrl + v	
8	ラベリングを全て選択	Ctrl + a	
9	ラベリングを削除	Delete	マウスカーソルをラベリングページに移動して、ラベリングアイテムを選択。
10	ラベリングツール—楕円形	l	
11	ラベリングツール—ポリゴン	g	
12	ラベリングツール—長方形	r	
13	ラベリングツール—ペン	b	
14	ラベリングツール—輪郭自動塗りつぶし	a	
15	ラベリングツール—消しゴム	e	
16	ラベリングツール—マスク	m	
17	ラベリングツール—選択	s	
18	現在のページのラベリングをクリア	Ctrl + l	
19	データセット画像を削除	Delete	カーソルをデータセットページに移動します。
20	データセットなどのリストを切り替える/スライダーをドラッグ	↑ ↓ →←	カーソルは対応するコントロールウィンドウページに移動します (デフォルトのショートカットキー)。

FAQ

- 1. カメラの露出を調整し、または補光を行って環境光の変化をシミュレートしてデータを取得することはできますか。**

いいえ。現場の照明が変化する場合は、異なる照明条件でデータを取得しなければなりません。手動で作られた照明環境で取得するデータは参考できません。
- 2. カメラの位置が固定されており、ワークの位置姿勢が変化する場合、カメラの位置を移動してワークの位置変化をシミュレートすることはできますか。**

いいえ。データを取得する前に、カメラの位置を固定にしなければなりません。カメラを移動すれば、ディープラーニングモデルの効果とカメラの外部パラメータに影響を与えます。カメラの位置が固定されており、ワークの位置姿勢が変化する場合、トレーニング中に ROI を適切に大きくすることができます。
- 3. 元のカメラで取得されたデータの品質が悪くてカメラを交換した後、元のデータの反復モデルを追加する必要はありますか。**

いいえ。カメラ交換後のデータを再取得して、モデルトレーニングを行う必要があります。
- 4. 背景を変更すると、ディープラーニングの効果に影響しますか。**

はい。背景を変更すると、モデルの認識に誤りや漏れが発生しますので、初期段階で背景を確認したらそれを変更しないでください。
- 5. 異なるカメラ型番と取り付け高さで取得したデータを同じモデルに使用してトレーニングできますか。**

はい。ROI にご注意ください。差異を小さくするために、異なる高さで ROI をそれぞれ選択してください。
- 6. 高反射性の金属部品を対処する場合、データ取得中に注意すべきことはありますか。**

露出過度/不足を回避してください。一部の露出過度を避けられない場合に、ワークの輪郭がはっきり認識できるようにしてください。
- 7. モデルの効果が良くない場合は、トラブルシューティングするにはどうすればよいですか。**

トレーニングデータの数量と品質、データの多様性、ROI パラメータ、照明などからトラブルシューティングします。

 - a. 数量：トレーニングデータの数量が、良好なモデル結果を取得するのに十分であるかどうかを確認します。
 - b. 品質：データ品質が最良かつ明確で、露出過度/不足があるかどうかを確認します。
 - c. データの多様性：データに現場で起こりうる全ての状況が含まれるかどうかを確認します。
 - d. ROI パラメータ：現場で設定されたパラメータはトレーニング中のデータと一致しているかどうかを確認します。
 - e. 照明：照明が変化するかどうか、また取得時と一致しているかどうかを確認します。
- 8. 現場で照明が複雑で、影や遮蔽などによる認識効果の不安定をどう改善しますか。**

実際状況に応じて、遮光や補光を行えば改善できます。

9. 現場データとトレーニングデータの ROI が一致しなければ、インスタンスセグメンテーションの信頼度のしきい値に影響するのはなぜですか。

トレーニングデータの ROI と一致しなければ、対象物がモデルの最適認識範囲内にないため、信頼度のしきい値に影響します。したがって、応用時の ROI をトレーニングデータの ROI と一致させる必要があります。

10. 紙箱のスーパーモデルは、どのようなシーンに適していますか。

同じ色やパターンの紙箱、または多種類の紙箱のデパレタイジング・パレタイジングに適しています。ただ、このモデルは紙箱が同じ段に水平に配置されたシーンにのみ適用されます。

11. 紙箱のスーパーモデルを使用すれば、データをどう取得しますか。

スーパーモデルを使用してテストします。正確に分割できなければ、問題のあるデータに対して約 20 枚の画像を取得します。具体的な方法は、[紙箱のデパレタイジング・パレタイジングのプロジェクト](#)をご参照ください。

12. 分類器は GPU なしで画像分類モデルを使用できますか？

いいえ。